SEQUENTIAL LOCALIZATION OF SENSOR NETWORKS

Fang, J.; Cao, M.; Morse, A. S.; Anderson, B. D. O.

Published in:
SIAM Journal on Control and Optimization

DOI:
10.1137/070679144

SIAM J. Control Optim.
Vol. 48, No. 1, pp. 321–350

SEQUENTIAL LOCALIZATION OF SENSOR NETWORKS[*]

J. FANG[†], M. CAO[‡], A. S. MORSE[†], and B. D. O. ANDERSON[§]

Abstract. The sensor network localization problem with distance information is to determine the positions of all sensors in a network, given the positions of some of the sensors and the distances between some pairs of sensors. A definition is given for a sensor network in the plane to be "sequentially localizable." It is shown that the graph of a sequentially localizable network must have a "bilateration ordering," and a polynomial time algorithm is given for deciding whether or not a network's graph has such an ordering. A provably correct algorithm is given which consists of solving a sequence of quadratic equations, and it is shown that the algorithm can localize any localizable network in the plane whose graph has a bilateration ordering.

Key words. sensor networks, localization, graph theory, rigidity

AMS subject classifications. 68W01, 68R99

DOI. 10.1137/070679144

1. Introduction.

Determining the positions of sensors is essential in many network applications such as geographic routing, coverage and creating formations. Equipping each sensor in a network with GPS is not feasible in many cases because of the large number of sensors and the cost associated with a GPS unit. Hence, we attack this problem by exploiting the connectivity of a sensor network and some common capabilities of sensors. More specifically, we assume a sensor can measure its distances to and communicate with certain other sensors in the network. The sensor network localization problem with distance information is to determine the positions of all sensors in a network given the positions of some sensors and the distances between some pairs of sensors. A sensor whose position is given is called an *anchor*. A network in $\mathbb{R}^d$ is said to be *localizable* if there exists exactly one position in $\mathbb{R}^d$ corresponding to each nonanchor sensor such that the given intersensor distances are satisfied. The authors of [5] use rigidity theory to give the necessary and sufficient conditions for a network to be localizable. However, the process of localizing a network has been shown to be NP-hard even when the network is known to be localizable [2]. This leaves us with the more refined questions of how we should go about localizing networks, and what kinds of networks can we "efficiently" localize.

The characterization of networks which can be "easily" or "efficiently" localized is not complete, even for the ideal case where the given distance measurements are exact. In [14], global nonlinear optimization techniques combined with heuristics are

[*] Received by the editors January 3, 2007; accepted for publication (in revised form) September 2, 2008; published electronically February 11, 2009. The work of Fang, Cao, and Morse was supported in part by grants from the U.S. Army Research Office and the U.S. National Science Foundation and by a gift from the Xerox Corporation. The work of Anderson was supported by the Research School of Information Sciences and Engineering and National ICT Australia which is funded by the Australian Government's Department of Communications, Information Technology and the Arts and the Australian Research Council through the Backing Australia's Ability initiative and the ICT Center of Excellence Program.

http://www.siam.org/journals/sicon/48-1/67914.html

[†] Department of Electrical Engineering, Yale University, New Haven, CT 06520 (jia.fang@yale.edu, as.morse@yale.edu).

[‡] Faculty of Mathematics and Natural Sciences, ITM, University of Groningen, Groningen, The Netherlands (m.cao@rug.nl).

[§] Australian National University and National ICT Australia Ltd., 216 Northbourne Ave., Canberra ACT 2601 Australia (Brian.Anderson@nicta.com.au).

used to estimate sensor information. In [13], a "fold-free" layout of the network is first estimated, and then force-based relaxation methods are used to refine the estimated sensor positions. In [16, 15], the distance between each pair of sensors is estimated from the given intersensor distances using a shortest path algorithm, and classical multidimensional scaling techniques are used to assign positions to each sensor to approximate the given distance information. In [3], a semidefinite programming–based algorithm is given for a class of dense networks when a sufficiently large number of intersensor distances are known. In this work, we are interested in provably correct localization algorithms and the kinds of networks that can be "efficiently" localized by them. We assume the given intersensor distances are exact distance measurements.

The characterization of efficiently localizable networks has been investigated in [1] and we extend the results of that paper. We present a localization algorithm called "Sweeps" which consists of solving a sequence of a finite number of quadratic equations, where the solution of each equation is easily obtainable by the well-known quadratic formula. We give a simple graphical characterization of all networks which can be localized by Sweeps, and we use graph rigidity theory to give some graphical characterizations of networks that can be efficiently localized by Sweeps. We also introduce the concept of "sequential" localization algorithms, and we say a network is *sequentially localizable* if it can be localized by some sequential localization algorithm. We show that Sweeps is a sequential localization algorithm which can localize *all* sequentially localizable networks. The Sweeps algorithm we present in this work is limited because we assume that the given intersensor distance measurements are exact. We refer the interested reader to [6] in which we adapt the Sweeps algorithm for the case of inaccurate distance measurements.

In section 2, we review the theoretical background of the localization problem from graph rigidity theory, and we give the terms and definitions to be used in the exposition that follows. In section 3, we introduce the notions of "bilateration orderings" and "sequentially localizable" networks. In section 4 we present the Restricted Sweeps algorithm on which the Sweeps algorithm is based, and in section 5 we present the Sweeps algorithm. In section 6 we characterize the class of networks localizable by Sweeps, and we show that all sequentially localizable networks are localizable by Sweeps. In section 7, we characterize some classes of networks which can be "efficiently" localized by either Sweeps or Restricted Sweeps, and in section 8 we characterize some classes of networks which are localizable by Restricted Sweeps. We conclude with future work and research problems in section 9.

2. Background.

A *multipoint* $p = \{p_1, \dots, p_n\}$ in d-dimensional space is a set of n points in $\mathbb{R}^d$ labeled $p_1, \dots, p_n$. Because we are concerned only with networks in the plane, we will henceforth restrict our attention to the case of $d = 2$. Two multipoints $p = \{p_1, \dots, p_n\}$ and $q = \{q_1, \dots, q_n\}$ of n points in $\mathbb{R}^2$ are *congruent* if for all $i, j \in \{1, \dots, n\}$, the distance between p_i and p_j is equal to the distance between q_i and q_j. A graph with vertex set $\mathcal{V}$ and edge set $\mathcal{E}$ is denoted $(\mathcal{V}, \mathcal{E})$. A *simple graph* is a graph for which there is at most one edge between any two distinct vertices, and no edge between a vertex and itself. A *point formation* of n points at a multipoint $p = \{p_1, \dots, p_n\}$ consists of p and a simple undirected graph $\mathbb{G}$ with vertex set $\mathcal{V} = \{1, \dots, n\}$, and is denoted by $(\mathbb{G}, p)$. If (i, j) is an edge in $\mathbb{G}$, then the *length of edge* (i, j) in the point formation $(\mathbb{G}, p)$ is the distance between p_i and p_j. Two point formations with the same graph have the same edge lengths in the case when the length of each edge in the graph is the same in both point formations.

For $t \in \mathbb{R}^2$, let $\|t\|$ denote the Euclidean norm on $\mathbb{R}^2$. For any multipoint $p =$

$\{p_1, \dots, p_n\}$ in $\mathbb{R}^2$ and $\epsilon > 0$, let $\mathcal{B}_p(\epsilon)$ denote the set of all multipoints $q = \{q_1, \dots, q_n\}$ in $\mathbb{R}^2$, where $\|p_i - q_i\| < \epsilon$ for all $i \in \{1, \dots, n\}$. A point formation $(\mathbb{G}, p)$ is *rigid* in $\mathbb{R}^2$ if there exists $\epsilon > 0$ such that for all $q \in \mathcal{B}_p(\epsilon)$, p and q are congruent whenever $(\mathbb{G}, p)$ and $(\mathbb{G}, q)$ have the same edge lengths. Roughly speaking, a rigid point formation is one that cannot be continuously deformed without causing an edge length to change. A graph $\mathbb{G}$ is said to be *rigid* in $\mathbb{R}^2$ if there exists a multipoint p in $\mathbb{R}^2$ and $\epsilon > 0$ such that $(\mathbb{G}, q)$ is rigid in $\mathbb{R}^2$ for all $q \in \mathcal{B}_p(\epsilon)$. A set consisting of a finite number of elements from $\mathbb{R}$ is said to be *algebraically independent over the rationals* if its elements do not satisfy any nonzero multivariable polynomial equation with rational coefficients. A multipoint is said to be *generic* if the set consisting of the coordinates of its points is algebraically independent over the rationals. It is known that if a multipoint p is generic, then a point formation $(\mathbb{G}, p)$ is rigid if and only if $\mathbb{G}$ is rigid.

A point formation $(\mathbb{G}, p)$ in $\mathbb{R}^2$ is *globally rigid* in $\mathbb{R}^2$ if multipoints p and q are congruent whenever $(\mathbb{G}, p)$ and $(\mathbb{G}, q)$ have the same edge lengths. In other words, edge lengths of a globally rigid point formation uniquely determine all intervertex distances. A graph $\mathbb{G}$ is said to be *globally rigid* in $\mathbb{R}^2$ if there exist multipoint p in $\mathbb{R}^2$ and $\epsilon > 0$ such that $(\mathbb{G}, q)$ is globally rigid in $\mathbb{R}^2$ for all $q \in \mathcal{B}_p(\epsilon)$. It is known that if a multipoint p in $\mathbb{R}^2$ is generic, then the point formation $(\mathbb{G}, p)$ is globally rigid in $\mathbb{R}^2$ if and only if $\mathbb{G}$ is globally rigid in $\mathbb{R}^2$. For any integer $k > 1$, a graph is said to be *k connected* if there does not exist a set of $k - 1$ vertices whose removal disconnects the graph. It is known that a graph with four or more vertices is globally rigid in $\mathbb{R}^2$ if and only if the graph is three connected and there does not exist an edge of the graph whose removal results in a graph which is not rigid in $\mathbb{R}^2$ [10, 9, 4]. There are a number of polynomial time algorithms such as Pebble Game for determining if a graph is rigid in $\mathbb{R}^2$ [11]. Since the k connectedness of a graph can also be efficiently determined, it follows that the global rigidity of a graph in $\mathbb{R}^2$ can be efficiently determined.

A network with n sensors is modeled by a point formation $(\mathbb{G}, p)$, where each sensor corresponds to exactly one vertex of $\mathbb{G}$, and vice versa, with (i, j) being an edge of $\mathbb{G}$ if either i and j are both anchors or the distance between the corresponding sensors is known and $p = \{p_1, \dots, p_n\}$, where p_i is the position of the sensor corresponding to vertex i. We say that $\mathbb{G}$ is the graph of the network and p is the multipoint of the network. In this work we will be concerned only with networks in the plane. It is known that if the multipoint of a network in $\mathbb{R}^2$ is generic, then the network is localizable if and only if it has at least 3 noncollinear anchors and the graph of the network is globally rigid in $\mathbb{R}^2$ [5]. Since *almost all* multipoints are generic, we will, without loss of generality, restrict our attention to those networks with generic multipoints [4]. In particular, for networks in the plane, this implies no two sensors occupy the same point and no three sensors are collinear in the networks we consider. For such networks, the localizability of the network depends only on the number of its anchors and its graph. Because we are concerned only with networks in the plane, we will refer to graphs that are globally rigid, or rigid, in $\mathbb{R}^2$ as simply globally rigid, or rigid. To avoid trivial and degenerate cases, we will restrict our attention to networks containing four or more sensors.

In the following, let $\mathbb{N}$ be a localizable network of $n > 3$ sensors in the plane labeled 1 through n, and suppose the multipoint of the point formation modeling $\mathbb{N}$ is generic. Let $\mathbb{G} = (\mathcal{V}, \mathcal{E})$ be the graph of $\mathbb{N}$. Since the multipoint of $\mathbb{N}$ is assumed to be generic, we have that $\mathbb{N}$ is localizable if and only if $\mathbb{G}$ is globally rigid in $\mathbb{R}^2$ and $\mathbb{N}$ has at least three anchors. As noted previously, there are a number of efficient algorithms for determining if a graph is globally rigid in $\mathbb{R}^2$ [11, 9, 10]. Hence, it follows that the localizability of $\mathbb{N}$ can also be efficiently determined just by analyzing the graph of $\mathbb{N}$ and counting its anchors. Without loss of generality, suppose that for each $i \in \{1, 2, \ldots, n\}$, vertex i of $\mathbb{G}$ corresponds to sensor i and vice versa. For each $v \in \mathcal{V}$, let $\mathcal{N}(v)$ denote the set consisting of all vertices u where $(u, v) \in \mathcal{E}$, and for each $u \in \mathcal{N}(v)$ write d_{uv} for the distance between sensors u and v.

3. Sequentially localizable networks. Suppose $\mathcal{A}$ is a set of at least three sensors of $\mathbb{N}$ and the vertices corresponding to the sensors in $\mathcal{A}$ induce a complete graph in $\mathbb{G}$, i.e., the distances among *all* pairs of sensors in $\mathcal{A}$ are known. Suppose a position $\pi(a)$ is assigned to each sensor $a \in \mathcal{A}$ such that all known distances among the sensors of $\mathcal{A}$ are satisfied. Since $\mathbb{N}$ is localizable, it is straightforward to show that the positions assigned to the sensors in $\mathcal{A}$ determine a unique position for each of the sensors not in $\mathcal{A}$. In other words, there corresponds exactly one position $\pi(v)$ to each of the sensors $v \in \mathcal{V} - \mathcal{A}$ such that all known intersensor distances are satisfied, i.e., $\|\pi(v) - \pi(u)\| = d_{uv}$ for all $(u, v) \in \mathcal{E}$. We call $\pi(v)$, $v \in \mathcal{V}$, the position of sensor v *relative* to $\mathcal{A}$, and we call $\mathcal{A}$ the set of *proxy anchors* of $\mathbb{N}$. It is easy to show that if sensors labeled a_1, a_2, a_3 are three anchors whose positions are given either by GPS or manual configuration, then the given anchor positions and $\pi(a_1), \pi(a_2), \pi(a_3)$ can be used to compute a Euclidean transformation which maps each $\pi(v)$, $v \in \mathcal{V}$, to the actual position of sensor v.

Let $\mathcal{A}$ denote a set of at least three proxy anchors of $\mathbb{N}$, i.e., $\mathcal{A}$ is any set of three sensors for which all distances among the sensors are given, and each sensor in $\mathcal{A}$ has been assigned a position so that the given distances among them are satisfied. Let $\pi(u)$, $u \in \mathcal{A}$, denote the position assigned to sensor u, and let $\pi(v)$, $v \in \mathcal{V}$, denote the position of sensor v relative to $\mathcal{A}$. For each sensor v and a set $\mathcal{S}$ of points in the plane, we say that $\mathcal{S}$ is a *candidate positions set* of sensor v if $\pi(v) \in \mathcal{S}$. If a candidate positions set consists of a finite number of points, then the set is said to be *finite*. By a *sweep* of $\mathbb{N}$ is meant any sequence $v_1, \ldots, v_n$ obtained by relabeling the n sensors in any way. By a *predecessor* of a sensor in a sweep is meant any other sensor preceding it in the sweep such that the distance between the two sensors is known. The concatenation of a finite number of sweeps in a specific order is a *multiple sweep*. By a *sequential localization algorithm* is meant any localization algorithm which processes the sensors in a network, one by one, in a predetermined sequence in such a way so that the sequence is a multiple sweep and the data for each successive sensor $v \in \mathcal{V} - \mathcal{A}$ in the sequence are either the empty set or a finite candidate positions set for v computed using only the known distances between v and its predecessors, and previously determined data for v and the predecessors of v. The data for sensor $a \in \mathcal{A}$ is assumed to be the singleton candidate positions set consisting of just its assigned position. Clearly, the position of sensor v relative to $\mathcal{A}$ is computed if a candidate positions set consisting of just one element is computed for v. Suppose a singleton candidate positions set has been computed for each sensor of $\mathbb{N}$. If $\mathbb{N}$ has three anchors, then the given positions of the anchors and their computed positions relative to $\mathcal{A}$ can be used to obtain a Euclidean transformation which maps the computed position of each sensor v to the actual position of sensor v. Since $\mathbb{N}$ is localizable, it must have at least three anchors, so $\mathbb{N}$ can be localized by a sequential localization algorithm followed by a Euclidean transformation. For any localizable network, we say the network is *sequentially localizable* if it can be localized by a sequential localization algorithm followed by a Euclidean transformation. Furthermore, we say the network is sequentially localizable in k *sweeps* if the sequence in which the sensors are processed is a multiple sweep, which is the concatenation of k sweeps.

A graph has a *bilateration ordering* if its vertices can be ordered as $v_1, \ldots, v_n$

so that the subgraph induced by v_1, v_2, and v_3 is complete, and each v_i, $i > 3$, is adjacent to at least *two* distinct vertices v_j, $j < i$. As noted previously, to avoid degenerate cases, all networks considered below will be assumed to contain at least four sensors. The following is an easily shown property of the graphs of sequentially localizable networks.

LEMMA 1. *A network is sequentially localizable only if its graph has a bilateration ordering.*

All proofs, unless otherwise stated, are given in the appendix.

A simple and well-known example of a sequential localization algorithm is based on the *trilateration* operation, where the position of each sensor is determined using its distances to three sensors whose positions have already been determined. Trilateration can be applied to any localizable network in the plane possessing a special type of sweep called a "trilateration ordering," which means an ordering $v_1, \dots, v_n$ of the vertices of the network's graph so that the subgraph induced by v_1, v_2, v_3 is complete and each v_i with $i > 3$ is adjacent to at least *three* distinct vertices v_j, $j < i$ [1]. Clearly, a trilateration ordering of a graph is also a bilateration ordering, while a bilateration ordering is not necessarily a trilateration ordering. Graphs with trilateration orderings are known to be globally rigid in $\mathbb{R}^2$ [1]. Suppose $\mathbb{G}$ has a trilateration ordering $v_1, \dots, v_n$. Assign positions $p_{v_1}, p_{v_2}, p_{v_3}$ to sensors v_1, v_2, v_3, respectively, so that their intersensor distances are satisfied, and let v_1, v_2, v_3 be the proxy anchors of $\mathbb{N}$. As noted previously, the multipoint of $\mathbb{N}$ is assumed to be generic, which implies no three sensor positions are collinear. Hence, beginning with v_4, trilateration can be used to determine a *unique* position p_{v_i} for each sensor v_i, $i > 3$, using the given distances between v_i and its predecessors in the ordering, and the computed positions of its predecessors. It is easy to show that the computed position of each sensor is the position of the sensor relative to the proxy anchors. Moreover, the actual sensor positions can be obtained from the computed positions p_v, $v \in \mathcal{V}$, via a Euclidean transformation. Hence, the network can be localized by a sequence of trilateration operations followed by a Euclidean transformation and is therefore sequentially localizable in one sweep. Furthermore, it is straightforward to show that a network's graph must have a trilateration ordering if the network is sequentially localizable in one sweep.

LEMMA 2. *A localizable network is sequentially localizable in one sweep if and only if its graph has a trilateration ordering.*

From Lemma 2, we have that localizable networks whose graphs have trilateration orderings are sequentially localizable; however, as we will show below, the converse need not be true. The central aims of this paper are to explicitly characterize the class of sequentially localizable networks and to present a sequential localization algorithm, called Sweeps, which can localize all sequentially localizable networks. The main result of this paper is the following.

THEOREM 1. *A localizable network is sequentially localizable if and only if the graph of the network has a bilateration ordering. All sequentially localizable networks are localizable by Sweeps.*

The proof of Theorem 1 is given in section 6. Given a network whose multipoint is generic, it is known that the network is localizable if and only if it has three anchors and its graph is globally rigid in $\mathbb{R}^2$ [5]. From Theorem 1, it follows that if a network has three anchors and its graph is globally rigid and has a bilateration ordering, then the network is sequentially localizable and can be localized by Sweeps. In section 3.1, we give a polynomial time algorithm for determining if a graph has a bilateration ordering, and for identifying a bilateration ordering of the graph when the graph has at least one such ordering. As noted previously, there are polynomial time algorithms for determining if a graph is globally rigid in $\mathbb{R}^2$. Hence, it can be efficiently determined if a network is sequentially localizable, and therefore localizable by Sweeps, just by analyzing the graph of the network.

The graph $\mathbb{H}$ as shown in Figure 1(a) can be easily verified to be three connected since there do not exist two vertices whose removal disconnects the graph. Furthermore, it can be shown using the Pebble Game, for example, that there does not exist an edge of $\mathbb{H}$ whose removal results in a graph which is not rigid in $\mathbb{R}^2$ [11]. Hence, $\mathbb{H}$ is globally rigid in $\mathbb{R}^2$ [9, 10]. It follows then that any network with three anchors and a generic multipoint and whose graph is $\mathbb{H}$ must be localizable [5]. Furthermore, $\mathbb{H}$ has a bilateration ordering but no trilateration ordering. Hence, if $\mathbb{H}$ is the graph of a network with three anchors, then it follows from Theorem 1 that the network is sequentially localizable and can be localized by Sweeps. However, since $\mathbb{H}$ does not have a trilateration ordering, the network cannot be localized by only a sequence of trilateration operations followed by a Euclidean transformation. We note that not all localizable networks are sequentially localizable. For example, the graph in Figure 1(b) does not have a bilateration ordering since any bilateration ordering must begin with three vertices, all of which are in either $\{1, 2, 3, 4, 5\}$ or $\{6, 7, 8, 9, 10\}$, and no vertex in $\{1, 2, 3, 4, 5\}$ is adjacent to at least two vertices in $\{6, 7, 8, 9, 10\}$, and vice versa. Furthermore, it can be checked that the graph is also globally rigid in $\mathbb{R}^2$ [11, 9, 10]. So if the graph in Figure 1(b) is the graph of a network with three anchors, then the network is localizable but not sequentially localizable.

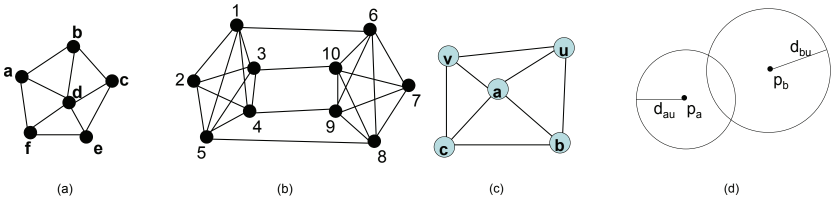

FIG. 1. (a) *A globally rigid graph with a bilateration ordering, i.e., a,d,f,e,c,b, but no trilateration ordering.* (b) *A globally rigid graph without a bilateration ordering.* (c), (d) *Sensors a, b, and c are anchors.*

3.1. Bilateration orderings. As noted in [1] a graph with a trilateration ordering must also be globally rigid in $\mathbb{R}^2$. It is easy to show by example that a graph with a bilateration ordering is not necessarily globally rigid in $\mathbb{R}^2$. However, a graph with a bilateration ordering is necessarily rigid in $\mathbb{R}^2$. More specifically, given any graph which is rigid in $\mathbb{R}^2$, it is known that if a new vertex x is added to the graph by making x adjacent to two or more vertices of the graph, then the resulting graph is again rigid in $\mathbb{R}^2$ [17]. Suppose $\mathbb{G}$ has a bilateration ordering, and let $v_1, v_2, v_3, \ldots, v_n$ be any bilateration ordering of $\mathbb{G}$. Let $\mathbb{G}_i$, $i \in \{3, 4, \ldots, n\}$, denote the graph induced in $\mathbb{G}$ by all vertices v_j where $j \leq i$. Since the complete graph on three vertices is rigid, it follows that $\mathbb{G}_3$ must be rigid. Now suppose $\mathbb{G}_i$ is rigid for some $i \in \{3, \ldots, n-1\}$. Since $\mathbb{G}_{i+1}$ can be obtained from $\mathbb{G}_i$ by making v_{i+1} adjacent to two or more vertices of $\mathbb{G}_i$, it follows that $\mathbb{G}_{i+1}$ must also be rigid. By induction then, $\mathbb{G}_n$, and therefore $\mathbb{G}$, must be rigid in $\mathbb{R}^2$. It is known that if a graph is rigid in $\mathbb{R}^2$, then the graph must also be two connected. Therefore, any graph with a bilateration ordering is rigid in $\mathbb{R}^2$ and two connected.

A graph may have zero, one, or multiple bilateration orderings. If there is no set of three vertices of $\mathbb{G}$ which induce a complete subgraph in $\mathbb{G}$, then $\mathbb{G}$ cannot have a bilateration ordering. Suppose $\mathbb{G}$ has at least one set of three vertices which induce a complete subgraph. In the following we give a polynomial time algorithm for determining if $\mathbb{G}$ has a bilateration ordering, and for identifying a bilateration ordering of $\mathbb{G}$ if there is at least one such ordering. Let x, y, z be any set of three vertices which induce a complete subgraph in $\mathbb{G}$, and let $\mathcal{W}_1 = \{x\}$, $\mathcal{W}_2 = \{y\}$, $\mathcal{W}_3 = \{z\}$. Suppose $\mathcal{W}_i$ for some $i \geq 3$ has been defined. If there exists a vertex u in $\mathcal{V} - \bigcup_{j \leq i} \mathcal{W}_j$ such that u is adjacent to at least two vertices in $\bigcup_{j \leq i} \mathcal{W}_j$, then define $\mathcal{W}_{i+1} = \{u\}$. Otherwise, set $\mathcal{W}_{i+1} = \emptyset$ and stop the algorithm. Let $\mathcal{W}_1, \dots, \mathcal{W}_h$ be the nonempty sets generated by this procedure. Clearly, $h = n$ if and only if there is a bilateration ordering of the graph beginning with x, y, z. If $h = n$, then the ordering obtained by labeling the vertex in $\mathcal{W}_i$, $i \in \{1, \dots, n\}$, as v_i is a bilateration ordering of $\mathbb{G}$. For each $i \in \{1, \dots, h\}$, we have that $|\mathcal{V} - \bigcup_{j \leq i} \mathcal{W}_j| = n - i$, and it takes a number of operations that are linear in n to check if a vertex in $\mathcal{V} - \bigcup_{j \leq i} \mathcal{W}_j$ is adjacent to two vertices in $\bigcup_{j \leq i} \mathcal{W}_j$. Hence, it takes a number of operations that are polynomial in n to determine each of the sets $\mathcal{W}_1, \dots, \mathcal{W}_h$, where h is at most n. The vertex labeling to obtain a bilateration ordering is clearly a linear time procedure. Furthermore, there are at most $\binom{n}{3}$ possible choices for the first three vertices of a bilateration ordering, which implies it can be determined in polynomial time if a graph has a bilateration ordering, and to identify a bilateration ordering if it exists. Since the global rigidity of a graph in $\mathbb{R}^2$ can also be efficiently determined, we conclude that it can be determined in polynomial time if a network is sequentially localizable just by analyzing the graph of the network.

In the following, we describe a class of graphs for which a bilateration ordering can be obtained beginning with *any* two adjacent vertices. A graph is called a *cycle* if its vertices can be relabeled as $c_1, \dots, c_m$, $m \geq 2$, such that c_i is adjacent to c_j if and only if $|i - j| = 1$ or $|i - j| = m - 1$. The *length* of a cycle is the number of edges in the cycle. Let $\mathbb{H}$ be any graph, and let $\mathbb{C}$ be any subgraph of $\mathbb{H}$ such that $\mathbb{C}$ is a cycle. If vertices u and v are adjacent in $\mathbb{H}$, and u and v are nonadjacent vertices in $\mathbb{C}$, then the edge (u, v) is called a *chord* of $\mathbb{C}$. A graph $\mathbb{H}$ is said to be *chordal* if for each subgraph which is also a cycle of length at least four, $\mathbb{H}$ contains at least one chord of the cycle. A chordal graph is not necessarily rigid.

LEMMA 3. *Let $\mathbb{H}$ be a rigid and chordal graph with at least four vertices. Then $\mathbb{H}$ has a bilateration ordering, and moreover, for each edge (u, v) in $\mathbb{H}$, there exists a bilateration ordering of $\mathbb{H}$ that begins with vertices u and v.*

Hence, if a graph is rigid and chordal, then the graph has a bilateration ordering, and furthermore, it is particularly easy to determine a bilateration ordering of the graph since any two adjacent vertices must be the first two vertices of some bilateration ordering. An additional simple consequence of Lemma 3 is that any rigid graph which is also chordal must contain a “triangle,” i.e., a cycle of length three.

4. The Restricted Sweeps algorithm.

In what follows we present a restricted version of the Sweeps algorithm, called Restricted Sweeps, for the class of networks whose graphs have bilateration orderings. The Sweeps algorithm is an extension of this and will be given in section 5.

We begin with an informal description of Restricted Sweeps. A bilateration ordering of the network's graph is first determined, assuming such an ordering exists, and the sensors corresponding to the first three vertices of the ordering are designated the proxy anchors. Positions are assigned to the proxy anchors so that the known distances among them are satisfied. For notational convenience, we assume each vertex of the network's graph has the same label as that of the sensor to which it corresponds. Roughly speaking, the algorithm "sweeps" through the network by processing the sensors sequentially according to the chosen bilateration ordering, beginning with the first sensor in the ordering which is not a proxy anchor. For each sensor which is not a proxy anchor, a finite candidate positions set of the sensor is computed using the known distances from the sensor to its predecessors in the ordering, and the candidate positions sets, or assigned positions, of its predecessors. Recall that a predecessor of a sensor in an ordering is simply any other sensor preceding it in the ordering such that the distance between the two sensors is known. Once the last sensor in the ordering is processed, a candidate positions set will have been computed for each sensor. We call this first "sweep" a *finite candidate positions set generating sweep.* If not every candidate positions set generated by the first sweep is a singleton, then subsequent "refining" sweeps are performed to remove, if possible, elements from each candidate positions set so as to obtain a candidate positions set of fewer elements. To perform a refining sweep, an ordering distinct from the one used to perform the previous sweep is determined, and the sensors are again processed sequentially according to the new ordering. In section 8, we will give a polynomial time algorithm for determining orderings so as to localize the network in as few sweeps as possible by analyzing the graph of the network. At the very least, the new ordering should be such that at least one sensor with a nonsingleton candidate positions set has a predecessor in the ordering. For each sensor v which is not a proxy anchor and whose candidate positions set is not a singleton, the candidate positions sets of v's predecessors in the new ordering, and the known distances between v and its predecessors, are used to identify, if possible, those points in v's candidate positions set which cannot be sensor v's position relative to the proxy anchors. The identified points are removed from the candidate positions set of sensor v to obtain a candidate positions set of fewer elements. We call each sweep after the first sweep a *refining* sweep since the goal of each subsequent sweep is to obtain smaller candidate positions sets.

To illustrate the general idea of a sweep, we use Restricted Sweeps to localize a simple network whose graph is shown in Figure 1(c). For each pair of adjacent vertices i, j, let d_{ij} denote the known distance between sensors labeled i and j. We assume the multipoint of the network is generic, which implies in particular that no three sensor positions are collinear. Vertices a, b, c correspond to the anchors, and vertices u, v correspond to sensors whose positions are to be determined. Let p_a, p_b, and p_c denote the positions of anchors a, b, and c, respectively. It can be efficiently determined that the graph in Figure 1(c) is globally rigid in $\mathbb{R}^2$ [9, 11, 10]. Since the network has three anchors and a generic multipoint, it follows that the network must be localizable.

For the sake of notational convenience, we choose for the first sweep the ordering a, b, c, u, v, in which case the proxy anchors correspond to the actual anchors. The algorithm begins by letting the candidate positions set of each anchor be the singleton set consisting of the anchor's position. Hence, the positions of sensors u and v relative to the anchors a, b, c are simply their actual positions. For a point $p \in \mathbb{R}^2$ and a positive real number r, let $\mathcal{C}(p, r)$ denote the circle with center p and radius r. Since u is the first nonanchor sensor in the chosen ordering, the algorithm proceeds in the first sweep by computing a finite candidate positions set for sensor u, which is just the set of points where the two circles $\mathcal{C}(p_a, d_{au})$ and $\mathcal{C}(p_b, d_{bu})$ intersect. Since no three sensor positions are collinear, it follows that $\mathcal{C}(p_a, d_{au})$ and $\mathcal{C}(p_b, d_{bu})$ must intersect at exactly two points. For instance, if $\mathcal{C}(p_a, d_{au})$ and $\mathcal{C}(p_b, d_{bu})$ are as shown in Figure 1(d), then the two points of intersection comprise the candidate positions set of sensor u. Let $\mathcal{S}(u,1)$ denote the candidate positions set computed for sensor u. Once a finite candidate positions set has been computed for sensor u, Restricted Sweeps proceeds in the first sweep by determining a finite candidate positions set for sensor v as follows. For each point $p \in \mathcal{S}(u,1)$ which is distinct from both p_a and p_c, let $\mathcal{I}(p)$ denote the points in the common intersection of the three circles $\mathcal{C}(p_a, d_{av})$, $\mathcal{C}(p_c, d_{cv})$, and $\mathcal{C}(p, d_{uv})$. The candidate positions set computed for sensor v, denoted $\mathcal{S}(v,1)$, is the union of all $\mathcal{I}(p)$, $p \in \mathcal{S}(u,1)$ where $p \neq p_a$ and $p \neq p_c$. Now we show that $\mathcal{S}(v,1)$ must be a singleton. Let p_v be any point in $\mathcal{S}(v,1)$. Clearly, the distance from p_v to anchors a and c must be d_{av} and d_{cv}, respectively. Furthermore, since p_v is in the intersection of $\mathcal{C}(p_a, d_{av})$, $\mathcal{C}(p_c, d_{cv})$, and $\mathcal{C}(p_u, d_{uv})$ for some $p_u \in \mathcal{S}(u,1)$, we have that the distance between p_u and p_v must be d_{uv}. Note that since $p_u \in \mathcal{S}(u,1)$, it follows that the distance between p_u and anchors a and b must be d_{ua} and d_{ub}, respectively. In other words, for each point $p_v \in \mathcal{S}(v,1)$, there exists a point $p_u \in \mathcal{S}(u,1)$ such that all known intersensor distances are satisfied when sensors v and u are assigned positions p_v and p_u, respectively, and the anchors are simply assigned their given positions. Since the network is localizable, we have that there exists exactly one point corresponding to each nonanchor sensor such that all known intersensor distances are satisfied. Hence, it must be the case that $\mathcal{S}(v,1)$ is a singleton. By definition, the point in $\mathcal{S}(v,1)$ must be the position of sensor v, so the first sweep not only computes a finite candidate positions set for sensor v, but also localizes sensor v since the computed candidate positions set is a singleton.

After the first sweep, a finite candidate positions set will have been determined for both u and v. Since the candidate positions set of sensor u, i.e., $\mathcal{S}(u,1)$, is not a singleton, a second ordering is determined in order to perform a refining sweep. Let the second ordering be a, b, c, v, u. Notice that the ordering has a sensor with a nonsingleton candidate positions set, namely u, that also has at least one predecessor in the ordering. More specifically, the predecessors of sensor u in the second ordering are sensors a, b, and v. The second sweep begins by considering the first vertex in the chosen ordering which has a nonsingleton candidate positions set, which in this case would be sensor u. The Restricted Sweeps algorithm identifies, and removes, points in $\mathcal{S}(u,1)$ which cannot be the position of sensor u as follows. The key observation is that if a point $p \in \mathcal{S}(u,1)$ is the position of sensor u, then for *each* of u's predecessors, there must exist a point in the candidate positions set of the predecessor such that p's distance to that point is the known distance between u and the predecessor. If this is not the case, then the point p can be removed from $\mathcal{S}(u,1)$ to obtain a new candidate positions set of fewer points.

Now we show that the second sweep will remove all but the actual position of sensor u from $\mathcal{S}(u,1)$. First, note that the distances between the actual position of sensor u and the points in the singleton candidate positions sets of a, b, and v must be d_{ua}, d_{ub}, and d_{uv}, respectively. So the actual position of sensor u will not be removed from $\mathcal{S}(u,1)$. Suppose there is a point $q \in \mathcal{S}(u,1)$ such that q is not removed by the second sweep; i.e., the distances between point q and the points in the singleton candidate positions sets of a, b, and v are d_{ua}, d_{ub}, and d_{uv}, respectively. Clearly, if sensor u is assigned point q as its position, and sensors a, b, c, v are assigned their actual positions, then all known intersensor distances are satisfied. Since the network is localizable, we have that there exists exactly one position corresponding to each nonanchor sensor such that all known intersensor distances are satisfied. This implies point q must be the actual position of sensor u, and all other points in $\mathcal{S}(u,1)$ will have been removed from $\mathcal{S}(u,1)$ by the second sweep. Hence, the second sweep localizes sensor u since the candidate positions set of u is a singleton after the second sweep.

Also, since the candidate positions set of each sensor is a singleton after the second sweep, it follows that Restricted Sweeps can localize the network in two sweeps.

4.1. Restricted Sweeps. Suppose the network $\mathbb{N}$ is localizable and the graph of $\mathbb{N}$, i.e., $\mathbb{G}$, has at least one bilateration ordering. We first give the terms and definitions to be used in describing the Restricted Sweeps algorithm. Let $2^{\mathbb{R}^2}$ be the power set of $\mathbb{R}^2$ and write $\mathbb{R}_+$ for the set of positive real numbers. Let $f : 2^{\mathbb{R}^2} \times \mathbb{R}_+ \to 2^{\mathbb{R}^2}$ denote the function $(\mathcal{S}, d) \longmapsto \mathcal{S}'$, where $\mathcal{S}'$ is the set of $p \in \mathbb{R}^2$ such that $\|p - t\| = d$ for some $t \in \mathcal{S}$. If $\mathcal{S} \in 2^{\mathbb{R}^2}$ is not empty, then geometrically $f(\mathcal{S}, d)$ is the union of all points in the plane which lie on circles with the same radius d centered at the points in $\mathcal{S}$. Of course if $\mathcal{S}$ is empty, then so is $f(\mathcal{S}, d)$ and conversely. We will be especially interested in the case when $\mathcal{S}$ is a nonempty "finite set" and $d > 0$, where by *finite set* we mean a set with a finite number of points in $\mathbb{R}^2$. In this case $f(\mathcal{S}, d)$ is simply the union of a finite number of circles in the plane which all have radius d.

Let $\mathbb{S}$ denote the set of all nonempty subsets of $\mathbb{R}^2$ with finitely many elements. Let q be a positive integer no smaller than 2 and write $\mathbb{S}^q$ for the q-fold Cartesian product of $\mathbb{S}$ with itself. Similarly, let $(\mathbb{R}_+)^q$ denote the q-fold Cartesian product of $\mathbb{R}_+$ with itself. Our goal is to define a function $g_q : \mathbb{S}^q \times (\mathbb{R}_+)^q \to 2^{\mathbb{R}^2}$ in such a way so that for each $\{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_q\} \in \mathbb{S}^q$ and $\{d_1, d_2, \dots, d_q\} \in (\mathbb{R}_+)^q$, $g_q(\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_q, d_1, d_2, \dots, d_q)$ is at most a finite set. Furthermore, we shall require the definition of g_q to be such that whenever there are distinct points $u_i \in \mathcal{S}_i$, $i \in \{1, 2, \dots, q\}$, if $v \in \mathbb{R}^2$ satisfies $\|v - u_i\| = d_i$, $i \in \{1, 2, \dots, q\}$, then v must be a point in $g_q(\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_q, d_1, d_2, \dots, d_q)$. Defining $g_q(\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_q, d_1, d_2, \dots, d_q)$ in the most obvious way as the intersection of the sets $f(\mathcal{S}_i, d_i)$, $i \in \{1, 2, \dots, q\}$, will not be adequate, for it may be the case that the resulting intersection is a continuous circle of points in the plane rather than a finite set. However, a necessary condition for this to occur is that $\bigcap_{j=1}^q \mathcal{S}_j \neq \emptyset$. Hence, let $\mathcal{I} = \bigcap_{j=1}^q \mathcal{S}_j$ and let $\mathcal{X}$ be the intersection of the sets $f(\mathcal{S}_1 \backslash \mathcal{I}, d_1)$ and $f(\mathcal{S}_i, d_i)$, $i \in \{2, \dots, q\}$, which is clearly finite. For each point p in $\mathcal{I}$, let $\mathcal{Y}(p)$ denote the intersection of $f(\{p\}, d_1)$ and $f(\mathcal{S}_i \backslash \{p\}, d_i)$, $i \in \{2, \dots, q\}$, which is again finite. By letting $g_q(\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_q, d_1, d_2, \dots, d_q)$ be the union of $\mathcal{X}$ and $\mathcal{Y}(p)$, $p \in \mathcal{I}$, it is easy to see that g_q satisfies all the aforementioned requirements.

More formally, for $\mathcal{S}_1, \dots, \mathcal{S}_q \in \mathbb{S}$, let $\mathcal{I} = \bigcap_{j=1}^q \mathcal{S}_j$. Let k denote the number of elements in $\mathcal{I}$. If $\mathcal{I}$ is not the empty set, i.e., $k > 0$, then let $p_1, p_2, \dots, p_k$ denote the elements of $\mathcal{I}$. For any set $\mathcal{S} \in \mathbb{S}$, and any subset $\mathcal{T} \subseteq \mathcal{S}$, let $\mathcal{S} \backslash \mathcal{T}$ denote the complement of $\mathcal{T}$ in $\mathcal{S}$. Define the function $g_q : \mathbb{S}^q \times (\mathbb{R}_+)^q \to 2^{\mathbb{R}^2}$ as follows:

$$\begin{aligned} g_q(\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_q, d_1, d_2, \dots, d_q) &= \Big(f(\mathcal{S}_1 \backslash \mathcal{I}, d_1) \cap f(\mathcal{S}_2, d_2) \cap \cdots \cap f(\mathcal{S}_q, d_q) \Big) \\ &\bigcup \left(\bigcup_{i=1}^{k} f(\{p_i\}, d_1) \cap f(\mathcal{S}_2 \backslash \{p_i\}, d_2) \cap \dots\dots \cap f(\mathcal{S}_q \backslash \{p_i\}, d_q) \right). \end{aligned} \tag{1}$$

For $q \geq 2$, it is easy to show that g_q is defined such that for each $\{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_q\} \in \mathbb{S}^q$ and $\{d_1, d_2, \dots, d_q\} \in (\mathbb{R}_+)^q$, $g_q(\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_q, d_1, d_2, \dots, d_q)$ is at most a finite set. Furthermore, whenever there are distinct points $u_i \in \mathcal{S}_i$, $i \in \{1, 2, \dots, q\}$, if $v \in \mathbb{R}^2$ satisfies $\|v - u_i\| = d_i$, $i \in \{1, 2, \dots, q\}$, then v must be a point in $g_q(\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_q, d_1, d_2, \dots, d_q)$.

Let $[v] = v_1, v_2, v_3, \dots, v_n$ be a bilateration ordering of $\mathbb{G}$. We begin by assigning a point $\pi(i)$ in $\mathbb{R}^2$ to each v_i, $i \in \{1, 2, 3\}$, so that the given distances among the sensors corresponding to v_i, $i \in \{1, 2, 3\}$, are satisfied. Let the proxy anchors of $\mathbb{N}$ be v_1, v_2, v_3. For each v_i, $i > 3$, let $\pi(v_i)$ denote the position of sensor v_i relative to the proxy anchors. In the following, we will describe an iterative procedure for computing a sequence of candidate positions sets for each $v \in \mathcal{V}$, i.e., $\mathcal{S}(v, 1), \mathcal{S}(v, 2), \ldots, \mathcal{S}(v, i), \ldots$.

For $i \in \{4, \ldots, n\}$, let $\mathcal{M}(v_i) = \mathcal{N}(v_i) \cap \{v_1, v_2, \ldots, v_{i-1}\}$. We denote the cardinality of $\mathcal{M}(v_i)$ by q_i and the elements of $\mathcal{M}(v_i)$ by $u_{i1}, u_{i2}, \ldots, u_{iq_i}$. Clearly $q_i \geq 2$ for all $i \in \{4, \ldots, n\}$ since $[v]$ is a bilateration ordering. We define the sets $\mathcal{S}(v_i, 1)$, $i \in \{1, 2, \ldots, n\}$, as follows. For $i \in \{1, 2, 3\}$, let

$$\mathcal{S}(v_i, 1) = \{\pi(i)\}, \tag{2}$$

and for $i \in \{4, 5, \ldots, n\}$, let

$$\mathcal{S}(v_i, 1) = g_{q_i}(\mathcal{S}(u_{i1}, 1), \mathcal{S}(u_{i2}, 1), \ldots, \mathcal{S}(u_{iq_i}, 1), d_{u_{i1}v_i}, d_{u_{i2}v_i}, \ldots, d_{u_{iq_i}v_i}). \tag{3}$$

Suppose $\mathcal{S}(v, k)$, $v \in \mathcal{V}$, have been computed. The sets $\mathcal{S}(v, k+1)$, $v \in \mathcal{V}$, are computed as follows. Let $[x] = x_1, x_2, \ldots, x_n$ be an ordering of $\mathcal{V}$, and for $i \in \{1, \ldots, n\}$ let $\mathcal{M}(x_i) = \mathcal{N}(x_i) \cap \{x_1, x_2, \ldots, x_{i-1}\}$. Note that $[x]$ need not be a bilateration ordering. For $i \in \{1, 2, 3, \ldots, n\}$, if $\mathcal{M}(x_i) = \emptyset$ or $|\mathcal{S}(x_i, k)| = 1$, then let

$$\mathcal{S}(x_i, k+1) = \mathcal{S}(x_i, k). \tag{4}$$

Otherwise, let

$$\mathcal{S}(x_i, k+1) = \mathcal{S}(x_i, k) \bigcap \left(\bigcap_{w \in \mathcal{M}(x_i)} f(\mathcal{S}(w, k+1), d_{wx_i}) \right). \tag{5}$$

4.2. Properties of the Restricted Sweeps algorithm. In the following, we will show that for all $v \in \mathcal{V}$, each $\mathcal{S}(v, i)$ is a finite candidate positions set for v, i.e., $\pi(v) \in \mathcal{S}(v, i)$, and $\mathcal{S}(v, j) \subseteq \mathcal{S}(v, i)$ if $i < j$.

Since $[v]$ is assumed to be a bilateration ordering, each $\mathcal{M}(v_i)$, $i > 3$, has at least two elements and so $q_i \geq 2$. Hence, for $i \in \{4, \ldots, n\}$, g_{q_i} is defined, and (3) implies that $\mathcal{S}(v_i, 1)$ is a finite set because the image of g_{q_i} consists of only finite sets. Since $\mathcal{S}(v_i, 1)$, $i \in \{1, 2, 3\}$, are also finite sets because of (2), we have that $\mathcal{S}(v, 1)$ is a finite set for each $v \in \mathcal{V}$. Note also that $\pi(v_i) \in \mathcal{S}(v_i, 1)$, $v_i \in \mathcal{V}$. This is clearly true for $i \in \{1, 2, 3\}$ because of (2). For any vertex $v \in \mathcal{V}$, an easily verified property of the function f is that if $u \in \mathcal{N}(v)$, and $\mathcal{S}(u)$ is a set for which $\pi(u) \in \mathcal{S}(u)$, then $\pi(v) \in f(\mathcal{S}(u), d_{uv})$. We call this the *position keeping* property of f. The fact that $\pi(v)$, $v \in \mathcal{V}$, are distinct, together with the definition of g_{q_i} and the position keeping property of f, implies that $\pi(v_i) \in \mathcal{S}(v_i, 1)$ for $i \in \{4, \ldots, n\}$. So each $\mathcal{S}(v, 1)$, $v \in \mathcal{V}$, is a finite candidate positions set for sensor v, and we call the computation of $\mathcal{S}(v, 1)$, $v \in \mathcal{V}$, a *finite position generating sweep* of $\mathbb{N}$. Suppose for some $k \geq 1$ that $\mathcal{S}(v, k)$, $v \in \mathcal{V}$, is a finite candidate positions set for v, i.e., $\pi(v) \in \mathcal{S}(v, k)$. For each x_i, (5) and (4) imply that $\mathcal{S}(x_i, k+1)$ must be a finite set since $\mathcal{S}(x_i, k)$ is a finite set. The fact that $\pi(x_i) \in \mathcal{S}(x_i, k)$ and the position keeping property of f imply $\pi(x_i) \in \mathcal{S}(x_i, k+1)$ for each x_i. So for each $v \in \mathcal{V}$, $\mathcal{S}(v, k+1)$ is a finite candidate positions set for v; furthermore, it is obvious from (5) and (4) that $\mathcal{S}(v, k+1) \subseteq \mathcal{S}(v, k)$ for all $v \in \mathcal{V}$. It follows from (2), (3), (5), and (4) that each $\mathcal{S}(v, i)$, $v \in \mathcal{V}$, $i \in \{1, \ldots, k+1\}$, is computed using $\mathcal{S}(u, i)$, where u is a predecessor of v in the ordering chosen for the ith sweep, and $\mathcal{S}(v, i-1)$ when $i > 1$. By definition then, the Restricted Sweeps algorithm is a sequential localization algorithm.

The preceding shows that if we sweep through the network a finite number of times beginning with a finite position generating sweep, we can generate a sequence of finite candidate positions sets for each $v \in \mathcal{V}$, i.e., $\mathcal{S}(v,1), \mathcal{S}(v,2), \ldots, \mathcal{S}(v,i), \ldots$, such that $\mathcal{S}(v,1) \supseteq \mathcal{S}(v,2) \supseteq \cdots \supseteq \mathcal{S}(v,i) \supseteq \cdots$. As we will show in section 4.3, each $\mathcal{S}(v,1)$, $v \in \mathcal{V}$, is obtained by solving a sequence of a finite number of quadratic equations, and each $\mathcal{S}(v,i)$, $v \in \mathcal{V}$, $i > 1$, is obtained by computing the distance between a finite number of specified pairs of points. Thus if we can sweep through the network a finite number of times, say k, such that for all $v \in \mathcal{V}$, each $\mathcal{S}(v,k)$ will contain just one element, then that element must be $\pi(v)$. Since $\mathbb{N}$ is localizable, $\mathbb{N}$ must have at least three anchors, so the sensor positions can be obtained from $\pi(v)$, $v \in \mathcal{V}$, via a Euclidean transformation computed from the anchors. In this case, we say the network is localizable by the Restricted Sweeps algorithm *in k sweeps* followed by a Euclidean transformation. In section 8, we will give the graph properties of networks for which we can choose sweep orderings so that the first sweep is a finite position generating sweep and the network is localized in as few sweeps as possible. We will also describe the procedure by which we can efficiently determine the sweep orderings by analyzing the network's graph.

4.3. Quadratic equations. The localization of $\mathbb{N}$ can be formulated mathematically as a system of $|\mathcal{E}|$ simultaneous quadratic equations in $|\mathcal{V}|$ variables:

$$(x_i - x_j)^2 + (y_i - y_j)^2 = d_{ij}^2 \quad \forall\ (i,j) \in \mathcal{E}, \tag{6}$$

where (x_i, y_i) denotes the unknown position of sensor i. In the following we show that the Restricted Sweeps algorithm is equivalent to solving a sequence of a finite number of quadratic equations, where each equation has just one unknown, the solution of which is easily obtained by the well-known quadratic formula, and computing the distance between a finite number of specified pairs of points.

We first consider the computation of $\mathcal{S}(v,1)$, $v \in \mathcal{V}$. Each $\mathcal{S}(v,1)$ is defined using the function g_q defined in (1). Since the ordering used for the first sweep must be a bilateration ordering, it must be the case that q is at least 2 in (3). Computing $g_q(\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_q, d_1, d_2, \ldots, d_q)$ is equivalent to solving the following system of equations in variables x and y for each collection of q points (a_i, b_i), $i \in \{1, \ldots, q\}$, where each $(a_i, b_i) \in \mathcal{S}_i$ and not all q points are identical:

$$(x - a_i)^2 + (y - b_i)^2 = d_i^2, \quad i \in \{1, \ldots, q\}. \tag{7}$$

First, consider the case where $q = 2$. The equations in (7) become

$$(x - a_1)^2 + (y - b_1)^2 = d_1^2, \tag{8}$$

$$(x - a_2)^2 + (y - b_2)^2 = d_2^2. \tag{9}$$

Equations (8) and (9) are satisfied by the coordinates of the points of intersection, if any, of the two circles with radii d_1 and d_2, and centered at (a_1, b_1) and (a_2, b_2), respectively. Since (a_1, b_1) and (a_2, b_2) are assumed to be nonidentical, the coordinates of at most two points in the plane can satisfy (8) and (9). See Figure 2 for the three cases where the two circles intersect at two, one, and zero points respectively. Equations (8) and (9) can be rewritten as one quadratic equation in one variable in the obvious way. Since (a_i, b_i), $i \in \{1, \ldots, q\}$, are assumed to be distinct, it must be

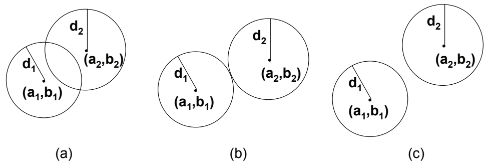

FIG. 2. (a) *Two intersection points.* (b) *One intersection point.* (c) *Zero intersection points.*

the case that either $a_1 - a_2 \neq 0$ or $b_1 - b_2 \neq 0$. Without loss of generality, suppose the latter is true. By subtracting (9) from (8), the quadratic terms cancel, and we get

$$y = \frac{d_1^2 - d_2^2 - (b_1^2 - b_2^2 + a_1^2 - a_2^2) - (-2a_1 + 2a_2)x}{-2b_1 + 2b_2}. \tag{10}$$

Hence, (8) can be rewritten as a quadratic equation of just the variable x:

$$(x - a_1)^2 + \left(\frac{d_1^2 - d_2^2 - (b_1^2 - b_2^2 + a_1^2 - a_2^2) - (-2a_1 + 2a_2)x}{-2b_1 + 2b_2} - b_1 \right)^2 = d_1^2. \tag{11}$$

Obviously, if (x, y) satisfies (8) and (9), then x must satisfy (11). On the other hand, suppose x satisfies (11) and y satisfies (10); then x and y must also satisfy (8). So, if we let $P_1(x, y) = (x - a_1)^2 + (y - b_1)^2 - d_1^2$, $P_2(x, y) = (x - a_2)^2 + (y - b_2)^2 - d_2^2$, and $P_3(x, y) = P_1(x, y) - P_2(x, y)$, then x, y satisfy $P_1(x, y) = 0$ and $P_3(x, y) = 0$. Since $P_3(x, y) = P_1(x, y) - P_2(x, y)$, this implies $P_2(x, y) = 0$, which implies that (9) is also satisfied by x and y. Therefore, x and y satisfy (8) and (9) if and only if they also satisfy (11) and (10). Hence, for the case where $q = 2$, solving for x, y which satisfy (8) and (9) reduces to solving a quadratic equation in x and then solving for y via substitution. Since we are interested only in points in the real plane whose coordinates satisfy (8) and (9), any complex solutions to (8) and (9) are discarded. Clearly, when $q > 2$, solving for x, y which satisfy (7) can be similarly reduced to solving a quadratic equation in x and then solving for y via substitution. Furthermore, it is not difficult to show that when $q > 2$ the solution to (7) can be obtained by just solving a linear system of equations. Since each $\mathcal{S}_j$, $j \in \{1, \dots, q\}$, is a finite set, it follows that computing $\mathcal{S}(v, 1)$, $v \in \mathcal{V}$, in Restricted Sweeps is equivalent to solving a sequence of a finite number of polynomial equations, each in one variable and each with degree at most two, the solution of which is easily obtained by the quadratic formula.

Now consider the computation of $\mathcal{S}(v, k)$, $v \in \mathcal{V}$, for $k > 1$. Let $\mathcal{M}(v)$ denote the vertices adjacent to v which also precede v in the ordering chosen for the kth sweep. If $\mathcal{M}(v) = \emptyset$, then $\mathcal{S}(v, k) = \mathcal{S}(v, k - 1)$, so suppose $\mathcal{M}(v)$ is nonempty, and let $u_1, \dots, u_m$, $m \geq 1$, denote the elements of $\mathcal{M}(v)$. When $\mathcal{M}(v)$ is nonempty, $\mathcal{S}(v, k)$ is computed using (5). It follows from (5) that $\mathcal{S}(v, k)$ is obtained by removing all points p from $\mathcal{S}(v, k-1)$ for which there does not exist points $p_1 \in \mathcal{S}(u_1, k), \dots, p_m \in \mathcal{S}(u_m, k)$ such that $\|p - p_i\| = d_{u_i v}$ for each $i \in \{1, \dots, m\}$. Hence, (5) consists of computing the distances between pairs of points in $\mathcal{S}(v, k - 1)$ and $\mathcal{S}(u_i, k)$, $i \in \{1, \dots, m\}$. From this, we conclude that Restricted Sweeps is equivalent to solving a sequence of a finite number of quadratic equations, where the solution of each equation is easily obtained by the well-known quadratic formula, and computing the distance between a finite number of specified pairs of points.

5. The Sweeps algorithm. An extension to the Restricted Sweeps algorithm was proposed in [7]. The Sweeps algorithm to be presented in what follows is based upon the extension to the Restricted Sweeps algorithm which we now describe.

Like the Restricted Sweeps algorithm, Sweeps is a localization algorithm for the class of networks whose graphs have bilateration orderings. As in Restricted Sweeps, the Sweeps algorithm "sweeps" through the network according to a predetermined bilateration ordering of the sensors and computes a finite candidate positions set for each sensor using the candidate positions sets of its predecessors and known distances. The key difference in Sweeps is that a "subassignment" function is associated with each point in the candidate positions set computed for a sensor. We illustrate this using a simple example. We first define an *assignment* of $\mathbb{N}$ to be any function $\alpha : \mathcal{V} \rightarrow \mathbb{R}^2$. By a *subassignment* of $\mathbb{N}$ is meant any function that is the restriction of an assignment to a nonempty subset of $\mathcal{V}$. Suppose u, v, w, x is a subsequence of the ordering chosen for the first sweep, i.e., the finite candidate positions set generating sweep, and suppose v and w are each adjacent to both u and x, as shown in Figure 3(a). Let $\mathcal{S}(u, 1)$, $\mathcal{S}(v, 1)$, and $\mathcal{S}(w, 1)$ denote the candidate positions sets of u, v, w, respectively, computed in the first sweep by Restricted Sweeps. Since u is a predecessor of both v and w in the ordering, $\mathcal{S}(u, 1)$ is used in the computations of both $\mathcal{S}(v, 1)$ and $\mathcal{S}(w, 1)$. More specifically, suppose v has predecessors u and u'. From (3), we have that each point $p_v \in \mathcal{S}(v, 1)$ is obtained by computing the intersection of circles centered at distinct points p_u and $p_{u'}$ for some $p_u \in \mathcal{S}(u, 1)$ and $p_{u'} \in \mathcal{S}(u', 1)$. Hence, p_v can be considered a candidate position of sensor v under the assumption that sensors u and u' are positioned at p_u and $p_{u'}$, respectively. A graphical illustration of this is shown in Figure 3(b).

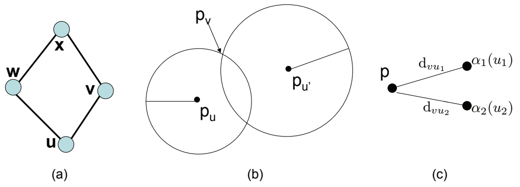

FIG. 3.

In the Restricted Sweeps algorithm, the candidate positions set of sensor v contains no "record" of the fact that p_v was computed assuming u is positioned at p_u and u' is positioned at $p_{u'}$. The Sweeps algorithm extends Restricted Sweeps by using a subassignment to keep track of the fact that p_v was computed assuming sensors u and u' are positioned at p_u and $p_{u'}$, respectively. So a subassignment β is associated with p_v, where the domain of β contains v, u, u' and $\beta(v) = p_v$, $\beta(u) = p_u$, and $\beta(u') = p_{u'}$. More generally, for each sensor v and each point p in the candidate positions set of v, the assumed position of each sensor whose candidate positions set was either directly or indirectly used in computing p is kept track of via a subassignment function. In reference to Figure 3(a), suppose $p_v \in \mathcal{S}(v, 1)$ is computed assuming sensor u is positioned at p_u, and $q_w \in \mathcal{S}(w, 1)$ is computed assuming sensor u is positioned at q_u, where $q_u \neq p_u$. Since both v and w are predecessors of x, we have that the candidate positions sets of both v and w are used in computing the candidate positions set of x. For the sake of this example, suppose that the only predecessors of x are v and w. In Restricted Sweeps, p_v and q_w would be used in computing the candidate positions set of x. More specifically, if the circle centered at p_v with radius d_{vx} and the circle centered at q_w with radius d_{wx} intersect at one or more points, then each of those points would be an element in the candidate positions set of x. In Sweeps, however, p_v and q_w would *not* be used in computing the candidate positions set of sensor x

because the two points were computed assuming different positions for sensor u. For certain networks, the candidate positions sets generated by Sweeps contain significantly fewer elements than those generated by Restricted Sweeps. And as we will see in section 7, the computational complexity of localizing a network by Sweeps, or Restricted Sweeps, is entirely dependent on the number of elements in the generated candidate positions sets.

5.1. Sweeps. Suppose the network $\mathbb{N}$ is localizable and the graph of $\mathbb{N}$, i.e., $\mathbb{G}$, has at least one bilateration ordering. We first give the terms and definitions to be used in describing the Sweeps algorithm. An assignment α is *consistent* if $\|\alpha(u)-\alpha(v)\| = d_{uv}$ for all $(u, v) \in \mathcal{E}$. Let $\mathcal{D}(\alpha)$ denote the domain of a subassignment α. Two subassignments α and β are said to be *consistent with each other*, and we write $\alpha \sim \beta$, if there does not exist $u \in \mathcal{D}(\alpha)\cap\mathcal{D}(\beta)$ such that $\alpha(u) \neq \beta(u)$. For $p \in \mathbb{R}^2$ and a positive real number r, let $\mathcal{C}(p, r)$ denote the circle of radius r centered at p. Let $\alpha_1, \dots, \alpha_k$ be a collection of $k \geq 1$ pairwise consistent subassignments, i.e., $\alpha_i \sim \alpha_j$ for all $i, j \in \{1, \dots, k\}$, and define $u_k(\alpha_1, \dots, \alpha_k)$ as the subassignment with domain $\bigcup_{i\in\{1,\dots,k\}} \mathcal{D}(\alpha_i)$ whose restriction to $\mathcal{D}(\alpha_i)$ is equal to α_i for each $i \in \{1, \dots, k\}$.

Consider a collection of $k \geq 2$ pairwise consistent subassignments $\alpha_1, \dots, \alpha_k$. Suppose there are vertices $v \in \mathcal{V}$ and $u_i \in \mathcal{D}(\alpha_i)$, $i \in \{1, \dots, k\}$, such that $(v, u_i) \in \mathcal{E}$ for all $i \in \{1, \dots, k\}$, and v is not an element of the domain of any α_i. If there is a point p whose distance to each $\alpha_i(u_i)$, $i \in \{1, \dots, k\}$, is d_{vu_i}, then roughly speaking, p is a candidate position for sensor v assuming each sensor u_i, $i \in \{1, \dots, k\}$, is positioned at $\alpha_i(u_i)$. More generally, p can be viewed as a candidate position for sensor v assuming each sensor $u \in \bigcup_{i\in\{1,\dots,k\}} \mathcal{D}(\alpha_i)$ is positioned at $\alpha(u)$, where $\alpha = u_k(\alpha_1, \dots, \alpha_k)$. We aim to define a set $\mathcal{M}(\alpha_1, \dots, \alpha_k, v, u_1, \dots, u_k)$ with the goal of keeping track of the candidate positions of sensor v assuming sensors u_i, $i \in \{1, \dots, k\}$, are positioned at $\alpha_i(u_i)$, $i \in \{1, \dots, k\}$, respectively. Since sensor positions are assumed to be distinct, we shall be interested only in the case where $\alpha_i(u_i)$, $i \in \{1, \dots, k\}$, are distinct. See Figure 3(c) for an illustration of the case when $k = 2$. To keep track of the fact that p is a candidate position for sensor v assuming each sensor u_i, $i \in \{1, \dots, k\}$, is positioned at $\alpha_i(u_i)$, define the subassignment β^p with domain $\{v\} \cup \bigcup_{i\in\{1,\dots,k\}} \mathcal{D}(\alpha_i)$ such that $\beta^p(v) = p$ and $\beta^p(u_i) = \alpha_i(u_i)$ for each $i \in \{1, \dots, k\}$:

$$\beta^p(v) = p, \quad \beta^p(u) = \zeta(u) \quad \forall\ u \in \bigcup_{i\in\{1,\dots,k\}} \mathcal{D}(\alpha_i), \tag{12}$$

where $\zeta = u_k(\alpha_1, \dots, \alpha_k)$. Let $\mathcal{M}(\alpha_1, \dots, \alpha_k, v, u_1, \dots, u_k)$ denote the set of all such β^p. More formally, if $\bigcap_{j\in\{1,\dots,k\}} \mathcal{C}(\alpha_j(u_j), d_{vu_j}) = \emptyset$, or $\alpha_i(u_i) = \alpha_j(u_j)$ for some $i, j \in \{1, \dots, k\}$, $i \neq j$, then let $\mathcal{M}(\alpha_1, \dots, \alpha_k, v, u_1, \dots, u_k) = \emptyset$. Otherwise, since $k \geq 2$, it is easy to see that $\bigcap_{j\in\{1,\dots,k\}} \mathcal{C}(\alpha_j(u_j), d_{vu_j})$ is a set consisting of at most q points in $\mathbb{R}^2$, where q is at most 2. Let the points be denoted by $p_1, \dots, p_q$, and let

$$\mathcal{M}(\alpha_1, \dots, \alpha_k, v, u_1, \dots, u_k) = \{\beta^{p_1}, \dots, \beta^{p_q}\}. \tag{13}$$

In the Sweeps algorithm, a sequence of finite sets of subassignments $\mathcal{S}(v, 1), \dots, \mathcal{S}(v, j)$ is computed for each $v \in \mathcal{V}$, where for each $i \in \{1, \dots, j\}$, v is in the domain of each subassignment in $\mathcal{S}(v, i)$, and $\{\beta(v) \mid \beta \in \mathcal{S}(v, i)\}$ is a finite candidate positions set for v.

Let $[v] = v_1, v_2, v_3, \dots, v_n$ be a bilateration ordering of $\mathbb{G}$. We begin by assigning a point $\pi(v_i)$ in $\mathbb{R}^2$ to each v_i, $i \in \{1, 2, 3\}$, so that the known distances among the sensors corresponding to v_i, $i \in \{1, 2, 3\}$, are satisfied. Let the proxy anchors of $\mathbb{N}$ be

v_1, v_2, v_3. For each v_i, $i > 3$, let $\pi(v_i)$ denote the position of sensor v_i relative to the proxy anchors. For v_i, $i \in \{1, 2, 3\}$, let α_i be the subassignment with domain $\{v_i\}$, where $\alpha_i(v_i) = \pi(v_i)$. For $i \in \{1, 2, 3\}$, let $\mathcal{S}(v_i, 1)$ be defined as

$$\mathcal{S}(v_i, 1) = \{\alpha_i\}, \quad i \in \{1, 2, 3\}. \tag{14}$$

The sets $\mathcal{S}(v_i, 1)$, $i > 3$, are computed iteratively as follows. For v_i, $i > 3$, let $\mathcal{M}(v_i) = \mathcal{N}(v_i) \cap \{v_1, \dots, v_{i-1}\}$. Since $[v]$ is a bilateration ordering, each $\mathcal{M}(v_i)$, $i > 3$, must be a set of at least two elements. Let $u_1, \dots, u_m$ be the elements of $\mathcal{M}(v_i)$. In order to compute $\mathcal{S}(v_i, 1)$, we consider each collection of pairwise consistent subassignments $\alpha_j \in \mathcal{S}(u_j, 1)$, $j \in \{1, \dots, m\}$. Suppose $\mathcal{M}(\alpha_1, \alpha_2, \dots, \alpha_m, v_i, u_1, u_2, \dots, u_m) \neq \emptyset$, and let $\beta \in \mathcal{M}(\alpha_1, \alpha_2, \dots, \alpha_m, v_i, u_1, u_2, \dots, u_m)$. From (12), we have that $v_i, u_1, \dots, u_m \in \mathcal{D}(\beta)$, $\beta(u_j) = \alpha_j(u_j)$ for all $j \in \{1, \dots, m\}$, and $\beta(v_i)$ is a candidate position of sensor v_i assuming that each sensor $u \in \mathcal{D}(\beta)$ is positioned at $\beta(u)$, and more specifically, that each sensor u_j, $j \in \{1, \dots, m\}$, is positioned at $\alpha_j(u_j)$. The set $\mathcal{S}(v_i, 1)$ is intended to be the set of all such subassignments β, where $\beta \in \mathcal{M}(\alpha_1, \alpha_2, \dots, \alpha_m, v_i, u_1, u_2, \dots, u_m)$ and $\alpha_j \in \mathcal{S}(u_j, 1)$, $j \in \{1, \dots, m\}$ is a collection of pairwise consistent subassignments. Hence, $\mathcal{S}(v_i, 1)$ is defined as

$$\mathcal{S}(v_i, 1) = \bigcup_{\alpha_j \in \mathcal{S}(u_j, 1)\ \forall\ j \in \{1, \dots, m\} \text{ and } \alpha_j \sim \alpha_k\ \forall\ j, k \in \{1, \dots, m\}} \mathcal{M}(\alpha_1, \dots, \alpha_m, v_i, u_1, \dots, u_m). \tag{15}$$

Note that since $|\mathcal{M}(v_i)| \geq 2$ for each v_i, where $i > 3$, it follows that each $\mathcal{S}(v, 1)$ consists of a finite number of elements.

Suppose for some $k \geq 1$ that $\mathcal{S}(u, k)$, $u \in \mathcal{V}$, have been computed, and that each $\mathcal{S}(u, k)$ consists of a finite number of elements. Let $u_1, \dots, u_n$, denoted $[u]$, be any ordering of the vertices such that the first three vertices of $[u]$ coincide with the proxy anchors: $u_1 = v_1$, $u_2 = v_2$, $u_3 = v_3$. Note that $[u]$ is not required to be a bilateration ordering. Once the ordering $[u]$ is selected, the sets $\mathcal{S}(u, k+1)$, $u \in \mathcal{V}$, are computed iteratively as follows. For $i \in \{1, 2, 3\}$, let $\mathcal{S}(u_i, k+1) = \mathcal{S}(u_i, k)$. For $i \in \{4, \dots, n\}$, let $\mathcal{M}(u_i) = \mathcal{N}(u_i) \cap \{u_1, \dots, u_{i-1}\}$, and let $\mathcal{S}(u_i, k+1) = \mathcal{S}(u_i, k)$ if $\mathcal{M}(u_i) = \emptyset$. If $\mathcal{M}(u_i)$ is nonempty, then let $w_1, \dots, w_m$ be the elements of $\mathcal{M}(u_i)$. For notational convenience, let w_0 denote u_i. Suppose α_0 is a subassignment in $\mathcal{S}(u_i, k)$ for which there exists a collection of subassignments $\alpha_j \in \mathcal{S}(w_j, k+1)$, $j \in \{1, \dots, m\}$, such that $\alpha_0, \alpha_1, \dots, \alpha_m$ are pairwise consistent and $\|\alpha_0(u_i) - \alpha_j(w_j)\| = d_{u_i w_j}$ for *all* w_j, $j \in \{1, \dots, m\}$. In this case, $\alpha_0(u_i)$ can be considered a candidate position for sensor u_i assuming that each w_j, $j \in \{1, \dots, m\}$, is positioned at $\alpha_j(w_j)$, and more generally, that each $w \in \mathcal{D}(\alpha_j)$, $j \in \{1, \dots, m\}$, is positioned at $\alpha_j(w)$. Hence, if α_0 is "augmented" to a subassignment α, where $\alpha = u_{m+1}(\alpha_0, \alpha_1, \dots, \alpha_m)$, then $\alpha(u_i) = \alpha_0(u_i)$ and $\alpha(w_j) = \alpha_j(w_j)$ for all $j \in \{1, \dots, m\}$, and $\alpha(w) = \alpha_j(w)$ for each $w \in \mathcal{D}(\alpha_j)$, $j \in \{1, \dots, m\}$. Roughly speaking, $\mathcal{S}(u_i, k+1)$ is the set of subassignments obtained from $\mathcal{S}(u_i, k)$ by "augmenting" each such subassignment α_0 to $u_{m+1}(\alpha_0, \alpha_1, \dots, \alpha_m)$. Now suppose β is a subassignment in $\mathcal{S}(u_i, k)$ for which there does *not* exist some collection of subassignments $\beta_j \in \mathcal{S}(w_j, k+1)$, $j \in \{1, \dots, m\}$, such that $\beta, \beta_1, \dots, \beta_m$ are pairwise consistent and $\|\beta(u_i) - \beta_j(w_j)\| = d_{u_i w_j}$ for *all* w_j, $j \in \{1, \dots, m\}$. It is straightforward to show that $\beta(u_i)$ cannot be the position of sensor u_i relative to the proxy anchors, and so β is not used to define any subassignment in $\mathcal{S}(u_i, k+1)$. Roughly speaking, $\beta(u_i)$ is removed from consideration as a candidate position for sensor u_i. More formally, $\mathcal{S}(u_i, k+1)$ is defined as

$$\begin{aligned}
&\text{(16)} \qquad \mathcal{S}(u_i, k+1) = \Big\{ u_{m+1}(\alpha_0, \alpha_1, \ldots, \alpha_m) \mid \alpha_0 \in \mathcal{S}(u_i, k), \\
&\qquad\qquad \alpha_j \in \mathcal{S}(w_j, k+1) \quad \forall\, j \in \{1, \ldots, m\}, \\
&\qquad\qquad \alpha_h \sim \alpha_j, \quad \alpha_h(w_h) \neq \alpha_j(w_j) \quad \forall\, h, j \in \{0, 1, \ldots, m\}, \\
&\text{(17)} \qquad\qquad \alpha_0(u_i) \in \bigcap_{j \in \{1, \ldots, m\}} \mathcal{C}(\alpha_j(w_j), d_{u_i w_j}) \Big\}.
\end{aligned}$$

Since each $\mathcal{S}(v, k)$, $v \in \mathcal{V}$, consists of a finite number of elements, it follows from (17) that $\mathcal{S}(v, k+1)$ must also consist of a finite number of elements.

By the same argument as that used in section 4.3, it follows that Sweeps is equivalent to solving a sequence of a finite number of quadratic equations, where each equation has just one unknown, the solution of which is easily obtained by the well-known quadratic formula, and computing the distance between a finite number of specified pairs of points.

5.2. Properties of Sweeps.

As noted previously, each of the sets computed by the Sweeps algorithm consists of a finite number of subassignments. In the following, we give some additional properties of these sets.

LEMMA 4. *Let w be any vertex of $\mathbb{G}$. If vertices u, v are adjacent in $\mathbb{G}$, and $u, v \in \mathcal{D}(\beta)$ for some $\beta \in \mathcal{S}(w, 1)$, then $\|\beta(u) - \beta(v)\| = d_{uv}$.*

From (17), we have that each subassignment $\alpha \in \mathcal{S}(v, 2)$, $v \in \mathcal{V}$, is a subassignment "augmented" from some subassignment $\hat{\alpha} \in \mathcal{S}(v, 1)$, i.e., $\alpha = u_{m+1}(\hat{\alpha}, \alpha_1, \ldots, \alpha_m)$. From this and Lemma 4 we can show the following.

LEMMA 5. *Let w be any vertex of $\mathbb{G}$. If vertices u, v are adjacent in $\mathbb{G}$, and $u, v \in \mathcal{D}(\beta)$ for some $\beta \in \mathcal{S}(w, 2)$, then $\|\beta(u) - \beta(v)\| = d_{uv}$.*

Recall that $v_1, \ldots, v_n$ was the ordering used to compute $\mathcal{S}(v, 1)$, $v \in \mathcal{V}$, and v_1, v_2, v_3 are the proxy anchors of $\mathbb{N}$ whose assigned positions are $\pi(v_1)$, $\pi(v_2)$, and $\pi(v_3)$, respectively. Since $\mathbb{N}$ is localizable, there exists exactly one consistent assignment $\bar{\alpha}$ of $\mathbb{N}$, where $\bar{\alpha}(v_i) = \pi(v_i)$ for each $i \in \{1, 2, 3\}$. Furthermore, for each $v \in \mathcal{V}$, $\bar{\alpha}(v)$ is the position of sensor v relative to the proxy anchors. Suppose $\mathcal{S}(v, 1), \mathcal{S}(v, 2), \ldots, \mathcal{S}(v, k)$ are computed for each $v \in \mathcal{V}$.

LEMMA 6. *For each $v \in \mathcal{V}$ and $i \in \{1, \ldots, k\}$, there is a $\beta \in \mathcal{S}(v, i)$ which is the restriction of $\bar{\alpha}$ to the domain of β and $v \in \mathcal{D}(\beta)$.*

Each $\mathcal{S}(v_i, k)$, $k \geq 1$, is computed using sets $\mathcal{S}(v_j, k)$, where v_j is a predecessor of v_i in the kth chosen ordering, and $\mathcal{S}(v_i, k-1)$ when $k > 1$. Recall that the sensors of $\mathbb{N}$ are labeled $1, \ldots, n$ and $\mathcal{V} = \{1, \ldots, n\}$. Each subassignment β may be represented as a sequence of n points, where the ith point in the sequence is $\beta(i)$ if $i \in \mathcal{D}(\beta)$ and is $\emptyset$ otherwise. Hence, if $\beta_1, \ldots, \beta_m$ are m subassignments, where subassignment β_i is represented by the sequence $p_{i1}, \ldots, p_{in}$, then the set consisting of subassignments $\beta_1, \ldots, \beta_m$ can be represented by the set of points $\{p_{11}, \ldots, p_{1n}, p_{21}, \ldots, p_{2n}, \ldots, p_{m1}, \ldots, p_{mn}\}$. From Lemma 6 we have that for each sensor v and each computed ith sweep, the set $\{\beta(v) \mid \beta \in \mathcal{S}(v, i)\}$ is a finite candidate positions set for v. By definition then, Sweeps is a sequential localization algorithm.

6. Graphical properties of networks localizable by Sweeps.

In the following, we show that the necessary condition for a localizable network to be sequentially localizable is also a sufficient condition for the network to be localizable by the Sweeps algorithm. More specifically, we show that all localizable networks whose graphs have bilateration orderings can be localized by computing $\mathcal{S}(v,k)$, $v \in \mathcal{V}$, where $k \leq 2$, with the Sweeps algorithm, and we give an efficient algorithm for determining the sensor ordering of each sweep.

Let $\mathcal{A}$ denote any set of three vertices in $\mathbb{G}$ which induce a complete graph in $\mathbb{G}$. Let $\mathbb{H}_1, \mathbb{H}_2, \dots, \mathbb{H}_c$ denote the maximally connected components of the subgraph of $\mathbb{G}$ induced by vertices in $\mathcal{V} - \mathcal{A}$. The following is a consequence of the assumption that $\mathbb{N}$ is localizable.

LEMMA 7. *For each $i \in \{1, \dots, c\}$, the graph induced in $\mathbb{G}$ by $\mathcal{A}$ and the vertices of $\mathbb{H}_i$ is globally rigid in $\mathbb{R}^2$.*

Let $\mathbb{H}$ be $\mathbb{H}_i$ for any $i \in \{1, \dots, c\}$, and let u be any vertex of $\mathbb{H}$. In the following we construct a partition of the vertex set of $\mathbb{H}$. Let $\mathcal{N}_0(u) = \{u\}$, and let $\mathcal{N}_1(u)$ denote the set of vertices in $\mathcal{V} - \mathcal{A}$ adjacent to u. Suppose for some integer $i \geq 1$, $\mathcal{N}_j(u)$, $j \in \{0, 1, \dots, i\}$, have been determined. Let $\mathcal{N}_{i+1}(u)$ denote the set of vertices $w \in \mathcal{V} - \mathcal{A}$, where $w \notin \bigcup_{j \in \{0,\dots,i\}} \mathcal{N}_j(u)$ and w is adjacent to a vertex in $\mathcal{N}_i(u)$. Since there are a finite number of vertices, there can be only a finite number of sets generated this way. Suppose we have $h+1$ sets generated this way: $\mathcal{N}_0(u), \mathcal{N}_1(u), \dots, \mathcal{N}_h(u)$. It is straightforward to show that the sets $\mathcal{N}_i(u)$, $i \in \{0, 1, \dots, h\}$, make up a partition of the vertices of $\mathbb{H}$. We call $\mathcal{N}_i(u)$, $i \in \{0, 1, \dots, h\}$, a *vertex partition* of $\mathbb{H}$. Let n' denote the number of vertices in $\mathbb{H}$. Select any n' elements of $\{|\mathcal{A}| + 1, \dots, n\}$, and order them as $i_1, i_2, \dots, i_{n'}$ so that $i_1 < i_2 < \dots < i_{n'}$. Assign indices 1 to $|\mathcal{A}|$ to vertices in $\mathcal{A}$ in any manner, and assign index $i_{n'}$ to vertex u. Assign the remaining indices i_j, $j \in \{1, 2, \dots, n'-1\}$, to vertices in $\mathcal{N}_1, \dots, \mathcal{N}_h$ beginning with $\mathcal{N}_1$ and $n'-1$; i.e., assign indices $i_{n'-1}$ to $i_{n'-|\mathcal{N}_1(u)|}$ to the vertices in $\mathcal{N}_1(u)$ in any manner, assign indices $i_{n'-|\mathcal{N}_1(u)|-1}$ to $i_{n'-|\mathcal{N}_1(u)|-|\mathcal{N}_2(u)|}$ to the vertices in $\mathcal{N}_2(u)$ in any manner, and so on. We call this ordering a *complete ordering* of the vertices of $\mathbb{H}$ *with respect to u and $\mathcal{A}$*, or just a *complete ordering* of the vertices of $\mathbb{H}$ *with respect to $\mathcal{A}$*.

For each $i \in \{1, \dots, c\}$, let u_i be any vertex in $\mathbb{H}_i$. Since the vertex sets of $\mathbb{H}_i$, $i \in \{1, \dots, c\}$, are pairwise disjoint, we can construct an ordering of $\mathcal{V}$ that is a complete ordering of $\mathbb{H}_i$ with respect to u_i and $\mathcal{A}$ for all $i \in \{1, \dots, c\}$. We call this a *complete ordering* of $\mathbb{G}$ *with respect to $u_1, \dots, u_c$ and $\mathcal{A}$*, or just a *complete ordering* of $\mathbb{G}$ *with respect to $\mathcal{A}$*.

Let $v_1, v_2, v_3, \dots, v_n$ be any bilateration ordering of $\mathbb{G}$, and suppose $\mathcal{S}(v,1)$, $v \in \mathcal{V}$, are computed using this ordering. This implies that sensors corresponding to v_1, v_2, v_3 make up the set of proxy anchors, and v_1, v_2, v_3 induce a complete subgraph in $\mathbb{G}$. Let $\mathcal{A} = \{v_1, v_2, v_3\}$, and let $\pi(v_1)$, $\pi(v_2)$, and $\pi(v_3)$ be the positions assigned to the proxy anchors v_1, v_2, v_3, respectively. Since $\mathbb{N}$ is localizable, there is exactly one consistent assignment $\bar{\alpha}$ of $\mathbb{N}$ such that $\bar{\alpha}(v_i) = \pi(v_i)$, $i \in \{1, 2, 3\}$. As noted previously, the actual sensor positions can be obtained from $\bar{\alpha}(v)$, $v \in \mathcal{V}$, via a Euclidean transformation computed using anchor positions. For $i \in \{1, \dots, c\}$, let u_i be any vertex in $\mathbb{H}_i$.

LEMMA 8. *Suppose the ordering used to compute the second sweep, i.e., $\mathcal{S}(v,2)$, $v \in \mathcal{V}$, is a complete ordering of $\mathbb{G}$ with respect to $u_1, \dots, u_c$ and $\{v_1, v_2, v_3\}$. For each $i \in \{1, \dots, c\}$, and all $\alpha \in \mathcal{S}(u_i, 2)$, $\mathcal{D}(\alpha)$ is the union of $\{v_1, v_2, v_3\}$ and the vertex set of $\mathbb{H}_i$.*

The following is a consequence of Lemmas 5, 6, 7, and 8.

LEMMA 9. *Suppose the ordering used to compute the second sweep, i.e., $\mathcal{S}(v,2)$, $v \in \mathcal{V}$, is a complete ordering of $\mathbb{G}$ with respect to $u_1, \dots, u_c$ and $\{v_1, v_2, v_3\}$. For each $i \in \{1, \dots, c\}$, $\mathcal{S}(u_i, 2)$ is a singleton, and the subassignment in $\mathcal{S}(u_i, 2)$ is the restriction of $\bar{\alpha}$ to the union of $\{v_1, v_2, v_3\}$ and the vertex set of $\mathbb{H}_i$.*

If the ordering used to compute the second sweep, i.e., $\mathcal{S}(v,2)$, $v \in \mathcal{V}$, is a complete ordering of $\mathbb{G}$ with respect to $u_1, \ldots, u_c$ and $\{v_1, v_2, v_3\}$, then Lemma 9 implies that each $\mathcal{S}(u_i, 2)$, $i \in \{1, \ldots, c\}$, consists of exactly one subassignment α_i, which is the restriction of $\bar{\alpha}$ to the union of $\mathcal{A}$ and the vertex set of $\mathbb{H}_i$. Each sensor v which is not a proxy anchor must correspond to a vertex in exactly one of the $\mathbb{H}_i$, $i \in \{1, \ldots, c\}$. If sensor v corresponds to a vertex in $\mathbb{H}_i$, then the position of sensor v relative to the proxy anchors, i.e., $\bar{\alpha}(v)$, is given by $\alpha_i(v)$.[1] We have just shown the following.

LEMMA 10. *If $\mathbb{N}$ is localizable and its graph has a bilateration ordering, then $\mathbb{N}$ can be localized by computing two sweeps of the Sweeps algorithm followed by a Euclidean transformation. The ordering of the first sweep is any bilateration ordering $v_1, v_2, v_3, \ldots, v_n$, and the ordering of the second sweep is a complete ordering of $\mathbb{G}$ with respect to $\{v_1, v_2, v_3\}$.*

Now we give the proof for Theorem 1. From Lemma 1, we have that a localizable network is sequentially localizable only if its graph has a bilateration ordering. Lemma 10 implies that Sweeps can localize all sequentially localizable networks since a sequentially localizable network's graph must have a bilateration ordering. Furthermore, since Sweeps is a sequential localization algorithm, Lemma 10 implies that a localizable network is sequentially localizable if its graph has a bilateration ordering. Hence, Lemmas 10 and 1 imply a localizable network is sequentially localizable if and only if its graph has a bilateration ordering.

In [7], it was shown via extensive simulations that Sweeps is practically feasible on uniformly random networks of 250 sensors with connectivity modeled by unit disk graphs despite having a worst case computational complexity that is exponential in the number of sensors. In section 7, we give the graph properties of some networks which can be efficiently localized using Sweeps.

7. Efficiently localizable networks. Consider a class of networks such that for each positive integer i, there is a network in the class with at least i sensors. We say that the class of networks is efficiently localizable by Sweeps (or Restricted Sweeps) if there is a constant c such that each network in the class can be localized by Sweeps (or Restricted Sweeps) in a number of operations that is at most n^c, where n is the number of the network's sensors. The computational complexity of localizing $\mathbb{N}$ by Sweeps, or Restricted Sweeps, is entirely dependent upon the number of elements in the sets $\mathcal{S}(v,1)$, $v \in \mathcal{V}$. More specifically, let $\mathcal{M}(v)$ denote the vertices preceding v and also adjacent to v in the ordering chosen for the first sweep. In both the Sweeps and Restricted Sweeps algorithm, the number of operations necessary to compute $\mathcal{S}(v,1)$, $v \in \mathcal{V}$, is equal to $C \prod_{u \in \mathcal{M}(v)} |\mathcal{S}(u,1)|$, where C is a constant that is independent of the number of sensors in $\mathbb{N}$. In the following, we give a graphical characterization for when a network is efficiently localizable by Sweeps and Restricted Sweeps. We emphasize that this is not a complete characterization of all such efficiently localizable networks. However, the general techniques used here can be used to determine additional efficiently localizable networks.

Suppose the graph of $\mathbb{N}$, namely $\mathbb{G}$, has a trilateration ordering, and that the ordering chosen for the first sweep is a trilateration ordering $v_1, \ldots, v_n$. It is easy to see that $\mathcal{S}(v,1)$ is a singleton and $\prod_{u \in \mathcal{M}(v)} |\mathcal{S}(u,1)| = 1$ for all $v \in \mathcal{V}$. Hence, the class of networks whose graphs have trilateration orderings is obviously efficiently localizable by Sweeps. The key property of the trilateration ordering which makes $\mathbb{N}$ efficiently localizable by Sweeps is that for *all* $i \in \{4, \ldots, n\}$, the graph induced in $\mathbb{G}$

[1] By a slight modification to the Sweeps algorithm, a singleton candidate positions set can be obtained for each sensor; however, we omit this step since it is unnecessary.

by vertex v_i, and all the vertices v_j, where $j < i$, is globally rigid. We now "relax" this property to define a "superbilateration" ordering. A graph with $n \geq 4$ vertices has a *superbilateration* ordering $v_1, v_2, v_3, \ldots, v_n$ if the graph contains a subgraph with the same vertex set which can be constructed inductively as follows beginning with the complete graph on three vertices labeled v_1, v_2, v_3. Suppose the graph being constructed already contains vertices $v_1, \ldots, v_i$, $i \geq 3$. If $i + 1$ is even, then v_{i+1} is added to the graph by making v_{i+1} adjacent to at least three vertices v_j where $j < i + 1$. Otherwise, if $i + 1$ is odd, then v_{i+1} can be added to the graph in one of two ways, the first of which is to make v_{i+1} adjacent to at least three vertices v_j where $j < i + 1$. Or, v_{i+1} can be added to the graph by making v_{i+1} adjacent to distinct vertices v_i, v_k, v_j, where v_k is adjacent to v_i, and removing the edge between v_i and v_k.

A 1-*extension* on a graph is the operation whereby two adjacent vertices of the graph are first selected, say vertices u and v, and a new vertex w is added to the graph by making w adjacent to vertices u, v, and x, where x is distinct from both u and v, and removing the edge between u and v [8]. An *edge-addition* on a graph is the operation whereby two nonadjacent vertices are made adjacent by insertion of a new edge. In [10], it was shown that the graph resulting from an edge-addition or 1-extension operation on any globally rigid graph of four or more vertices is again globally rigid. From this, it follows that any graph with a superbilateration ordering $v_1, \ldots, v_n$ is necessarily globally rigid. Furthermore, for each $i > 3$, where either i is equal to n or i is odd, the graph induced by all vertices v_j, $j \leq i$, is globally rigid. Clearly, a trilateration ordering is automatically a superbilateration ordering. It is easy to show by example that the converse need not be true. Suppose $\mathbb{G}$ has a superbilateration ordering $v_1, \ldots, v_n$. Let v_1, v_2, v_3 be the proxy anchors of $\mathbb{N}$. For $i > 3$, and where i is either odd or equal to n, let $\mathbb{N}_i$ denote the subnetwork consisting of all sensors corresponding to vertices v_j, $j \leq i$. Each subnetwork $\mathbb{N}_i$ can be efficiently localized, relative to the proxy anchors, by Sweeps assuming the positions of all sensors in $\mathbb{N}_i$ which are also in some $\mathbb{N}_j$, $j < i$, are known. Hence, the entire network can be localized in a number of operations polynomial in the number of sensors by using Sweeps to localize each of the subnetworks in sequence beginning with $\mathbb{N}_5$. Generally speaking, suppose a localizable network contains subnetworks $\mathbb{N}_1, \ldots, \mathbb{N}_m$ so that each subnetwork $\mathbb{N}_i$ is efficiently localizable by Sweeps (or Restricted Sweeps) assuming the position of each sensor in $\mathbb{N}_i$ which is also in some $\mathbb{N}_j$, $j < i$, is known. Then the entire network is efficiently localizable by localizing the subnetworks $\mathbb{N}_1, \ldots, \mathbb{N}_m$ in sequence, provided each sensor of $\mathbb{N}$ is in some $\mathbb{N}_i$, $i \in \{1, \ldots, m\}$.

8. Graphical properties of networks localizable by Restricted Sweeps.

In this section, we will give sufficient conditions on the graphs of localizable networks for which we can choose sweep orderings so that the network is localized in as few sweeps as possible by the Restricted Sweeps algorithm. First, consider the case where $\mathbb{N}$'s graph $\mathbb{G}$ has a trilateration ordering $v_1, \ldots, v_n$, and suppose this is the ordering chosen for the first sweep. Since we assume that the multipoints of the networks we consider are generic, it follows that no three sensor positions of $\mathbb{N}$ are collinear. Hence, each $\mathcal{S}(v, 1)$, $v \in \mathcal{V}$, as computed by the first sweep of the Restricted Sweeps algorithm, is a singleton. This and Lemma 2 imply that a network with three or more anchors can be localized by Restricted Sweeps in one sweep followed by a Euclidean transformation if and only if its graph has a trilateration ordering.

8.1. Networks with partially acyclic graphs.

In the following, we show that if $\mathbb{G}$ is "partially acyclic," then $\mathbb{G}$ must have a bilateration ordering and $\mathbb{N}$ is localizable by Restricted Sweeps in two sweeps plus a Euclidean transformation. For any subset $\mathcal{W}$ of $\mathcal{V}$, let $\mathbb{G}(\mathcal{W})$ denote the graph induced in $\mathbb{G}$ by vertices in $\mathcal{W}$. For any nonempty subset $\mathcal{W}$ of $\mathcal{V}$, we say that $\mathbb{G}$ is *partially acyclic with respect to* $\mathcal{W}$, or just *partially acyclic*, if $\mathbb{G}(\mathcal{W})$ is a complete graph and $\mathbb{G}(\mathcal{V}-\mathcal{W})$ is acyclic. Suppose $\mathbb{G}$ is partially acyclic with respect to $\mathcal{W}$, and that each vertex in $\mathcal{V}-\mathcal{W}$ has degree at least three in $\mathbb{G}$. In the following, we will construct a bilateration ordering of $\mathbb{G}$. We first note that a necessary condition for a graph with four or more vertices to be globally rigid in $\mathbb{R}^2$ is that each of its vertices must have degree at least three [9]. Since $\mathbb{N}$ is localizable and contains at least four sensors, it follows that each vertex of $\mathbb{G}$ must have degree at least three. The graph in Figure 1(a) is globally rigid and partially acyclic with respect to any three mutually adjacent vertices of the graph. Additional globally rigid graphs which are also partially acyclic can be constructed using the edge-addition and 1-extension operations beginning with the complete graph on four vertices [10].

Let $\mathbb{H}$ denote a maximally connected component of $\mathbb{G}(\mathcal{V}-\mathcal{W})$, and let r denote any vertex of $\mathbb{H}$. Let $[r]$ denote any complete ordering of the vertex set of $\mathbb{H}$ with respect to r and $\mathcal{W}$, and let $\mathcal{N}_0(r), \mathcal{N}_1(r), \dots, \mathcal{N}_h(r)$ denote the vertex partition used to construct the ordering $[r]$. We now show that $[r]$ is a bilateration ordering of the graph induced in $\mathbb{G}$ by the vertices of $\mathbb{H}$ and the vertices in $\mathcal{W}$. Let v be any vertex of $\mathbb{H}$ and suppose $v \in \mathcal{N}_i(r)$ for some $i \in \{0, 1, \dots, h\}$. First suppose $i = 0$, in which case v must equal r. Suppose r is adjacent to $c < 2$ vertices of $\mathcal{W}$. Since $\mathbb{H}$ is a maximally connected component of $\mathbb{G}(\mathcal{V}-\mathcal{W})$, and r has degree at least three in $\mathbb{G}$, we have that r must be adjacent to at least $3 - c > 0$ vertices in $\mathbb{H}$, which implies r must be adjacent to at least three vertices preceding it in the ordering $[r]$. Now suppose $i > 0$. This implies v is adjacent to at least one vertex in $\mathcal{N}_{i-1}(r)$. Moreover, v is adjacent to exactly one vertex in $\mathcal{N}_{i-1}(r)$, for if v is adjacent to two vertices in $\mathcal{N}_{i-1}(r)$, then $\mathbb{H}$ is not acyclic, which implies $\mathbb{G}(\mathcal{V}-\mathcal{W})$ is not acyclic. Similarly, if v is adjacent to a vertex in $\mathcal{N}_i(r)$, then that would again imply $\mathbb{H}$ is not acyclic. Since v has degree three in $\mathbb{G}$, v must be adjacent to at least two vertices in $\mathcal{N}_{i+1}(r) \cup \mathcal{W}$. Since the vertices in $\mathcal{N}_{i+1}(r) \cup \mathcal{W}$ all precede v in $[r]$, it follows that v is adjacent to at least two vertices preceding it in the ordering $[r]$. Now we show that the first three vertices of $[r]$ induce a complete graph. Let x be any vertex in $\mathcal{N}_h(r)$. Since $\mathbb{H}$ is acyclic, x can be adjacent to exactly one vertex in $\mathbb{H}$. Also, since x has degree at least three in $\mathbb{G}$, it follows that x must be adjacent to at least two vertices in $\mathcal{W}$ and so $|\mathcal{W}| \geq 2$. Furthermore, since the vertices in $\mathcal{N}_h(r)$ precede all other vertices in $\mathbb{H}$ in the ordering $[r]$, it follows that $v_3 \in \mathcal{W} \cup \mathcal{N}_h(r)$. Hence, the first three vertices of $[r]$ induce a complete graph, and $[r]$ must therefore be a bilateration ordering. Let $\mathcal{V}(\mathbb{H})$ denote the vertex set of $\mathbb{H}$. We have just shown the following.

LEMMA 11. *If $\mathbb{G}$ is partially acyclic with respect to some $\mathcal{W} \subseteq \mathcal{V}$, and each vertex in $\mathcal{V}-\mathcal{W}$ has degree at least three in $\mathbb{G}$, then any complete ordering of a maximally connected component $\mathbb{H}$ of $\mathbb{G}(\mathcal{V}-\mathcal{W})$ with respect to $\mathcal{W}$ is also a bilateration ordering of the graph $\mathbb{G}(\mathcal{V}(\mathbb{H}) \cup \mathcal{W})$.*

Let $\mathbb{H}_1, \dots, \mathbb{H}_c$ denote the maximally connected components of $\mathbb{G}(\mathcal{V}-\mathcal{W})$. For each $\mathbb{H}_i$, let $v_{i1}, v_{i2}, v_{i3}, \dots, v_{ik}$ be any complete ordering of $\mathbb{H}_i$ with respect to $\mathcal{W}$. This implies that the first $|\mathcal{W}|$ vertices of each of the orderings must be the vertices of $\mathcal{W}$, i.e., $\{v_{i1}, \dots, v_{i|\mathcal{W}|}\} = \mathcal{W}$ for all $i \in \{1, \dots, c\}$. Let $w_1, \dots, w_{|\mathcal{W}|}$ denote the vertices of $\mathcal{W}$. From Lemma 11, we have that each of the orderings $v_{i1}, v_{i2}, v_{i3}, \dots, v_{ik}$ is a bilateration ordering. Therefore, the ordering obtained by concatenating $w_1, \dots, w_{|\mathcal{W}|}$ and $v_{i(|\mathcal{W}|+1)}, \dots, v_{ik}$ for all $i \in \{1, \dots, c\}$, i.e., $w_1, \dots, w_{|\mathcal{W}|}, v_{1(|\mathcal{W}|+1)}, \dots, v_{1k}, \dots, v_{i(|\mathcal{W}|+1)}, \dots, v_{ik}, \dots, v_{c(|\mathcal{W}|+1)}, \dots, v_{ck}$ must be a bilateration ordering, and it is a *complete ordering* of $\mathbb{G}$ with respect to $\mathcal{W}$. We have just shown the following.

LEMMA 12. *If $\mathbb{G}$ is partially acyclic with respect to some $\mathcal{W} \subseteq \mathcal{V}$, and each vertex in $\mathcal{V} - \mathcal{W}$ has degree at least three in $\mathbb{G}$, then $\mathbb{G}$ has a bilateration ordering, and any complete ordering of $\mathbb{G}$ with respect to $\mathcal{W}$ is a bilateration ordering.*

Remark 1. It is known that a necessary condition for a graph with at least four vertices to be globally rigid in $\mathbb{R}^2$ is that the graph must be three connected, i.e., $\mathbb{G}(\mathcal{V} - \mathcal{V}')$ is connected if $|\mathcal{V}'| \leq 2$. Hence, if $\mathbb{G}$ is partially acyclic with respect to $\mathcal{W}$, and $\mathbb{G}(\mathcal{V} - \mathcal{W})$ has more than one connected component, then $|\mathcal{W}| \geq 3$. Since $\mathbb{G}$ is globally rigid, Lemma 7 can be used to show that for each $i \in \{1, \dots, c\}$, the graph $\mathbb{G}(\mathcal{W} \cup \mathcal{V}_i)$, where $\mathcal{V}_i$ is the vertex set of $\mathbb{H}_i$, must also be globally rigid.

Our main result for networks with partially acyclic graphs is the following.

THEOREM 2. *A localizable network with graph $\mathbb{G}$ is localizable by Restricted Sweeps in two sweeps plus a Euclidean transformation if $\mathbb{G}$ is partially acyclic with respect to some $\mathcal{W} \subseteq \mathcal{V}$. The ordering of the finite position generating sweep is $[v] = v_1, v_2, v_3, \dots, v_n$, where $[v]$ is a complete ordering of $\mathbb{G}$ with respect to $\mathcal{W}$, and the ordering of the second sweep is $v_1, v_2, v_3, v_n, v_{n-1}, \dots, v_4$.*

A globally rigid graph in $\mathbb{R}^2$ is said to be *minimally* globally rigid in $\mathbb{R}^2$ if no edge can be removed from the graph without causing the graph to no longer be globally rigid in $\mathbb{R}^2$. A number of globally rigid graphs in $\mathbb{R}^2$ that are partially acyclic with respect to some $\mathcal{W} \subseteq \mathcal{V}$, where $|\mathcal{W}| \geq 3$, are also minimally globally rigid in $\mathbb{R}^2$. Hence, Theorem 2 implies that Restricted Sweeps can localize certain networks with just enough edges in their graphs to ensure localizability. For any $i > 3$, let $\mathbb{W}_i$ denote the graph whose vertices can be labeled as $w_0, w_1, \dots, w_i$ such that w_0 is adjacent to all other vertices, and vertices $w_1, \dots, w_i$ induce a cycle in the graph. Any such $\mathbb{W}_i$, $i > 3$, is called a *wheel graph*. It is known that wheel graphs are minimally globally rigid, and it is straightforward to show that any wheel graph is partially acyclic with respect to any three vertices which are mutually adjacent. Hence, any network with three or more anchors and whose graph is a wheel graph is localizable by Restricted Sweeps in two sweeps plus a Euclidean transformation. One can show by example that globally rigid graphs which are also partially acyclic are not limited to just wheel graphs.

Let $\mathcal{N}_T$ denote the class of networks whose graphs have a trilateration ordering, and let $\mathcal{N}_P$ denote the class of networks whose graphs are globally rigid in $\mathbb{R}^2$ and partially acyclic. It is not difficult to show that $\mathcal{N}_T$ and $\mathcal{N}_P$ are not disjoint, $\mathcal{N}_T \nsubseteq \mathcal{N}_P$ and $\mathcal{N}_P \nsubseteq \mathcal{N}_T$. For example, networks with wheel graphs are in $\mathcal{N}_P$ but not $\mathcal{N}_T$, and any network whose graph has a trilateration ordering $v_1, \dots, v_n$, where $n > 5$ and each v_i, $i > 3$, is adjacent to vertices v_{i-1}, v_{i-2}, and v_{i-3}, is in $\mathcal{N}_T$ but not $\mathcal{N}_P$.

8.2. Networks with ring squared graphs. Many practical networks are such that the distance between two sensors is known if the sensors are within a prescribed sensing radius of each other. Suppose $\bar{\mathbb{N}}$ is such a network and has at least three anchors, and let $\bar{\mathbb{G}}$ be its graph. Define a *ring graph* with ordering $v_1, \dots, v_n$ as a graph whose vertices can be labeled as $v_1, \dots, v_n$ so that each vertex v_i, $1 < i < n$, is adjacent to vertices v_{i-1} and v_{i+1}, and vertex v_1 is adjacent to vertex v_n.

LEMMA 13. *If $\bar{\mathbb{G}}$ is a ring graph with ordering $v_1, \dots, v_n$, then $\bar{\mathbb{N}}$ is localizable in two sweeps plus a Euclidean transformation after doubling the sensing radius of each sensor. The ordering of the first sweep is $v_1, \dots, v_n$, and the ordering of the second sweep is $v_1, v_2, v_3, v_n, v_{n-1}, \dots, v_4$.*

Let $\bar{\mathcal{V}}$ and $\bar{\mathcal{E}}$ denote the vertex set and edge set of $\bar{\mathbb{G}}$, respectively. The *second power* of $\bar{\mathbb{G}}$, written $\bar{\mathbb{G}}^2$, is the graph with vertex set $\bar{\mathcal{V}}$ and edge set $\bar{\mathcal{E}} \cup \bar{\mathcal{E}}^2$, where $(i, j) \in \bar{\mathcal{E}}^2$ in the case when $i, j \in \bar{\mathcal{V}}$ and there exists $k \in \bar{\mathcal{V}}$ such that $(i, k), (k, j) \in \bar{\mathcal{E}}$.

A graph is *edge* 2*-connected* if there exists two paths with no edge in common between each pair of vertices. It is known that the second power of an edge 2-connected graph is globally rigid in $\mathbb{R}^2$ [1]. An important consequence of this and Theorem 13 is that if the graph of a network is edge 2-connected with at least three anchor vertices, and the network is such that the distance between two sensors is known if the sensors are within sensing radius, then the network is sequentially localizable after doubling the sensing radius of all the sensors [1].

9. Conclusion. In this work, we presented Sweeps, a sequential localization algorithm which consists of solving a sequence of a finite number of quadratic equations, and determining the distances between specified pairs of points. We identified the graph properties of all networks which can be localized by Sweeps, as well as the graph properties of some networks which can be efficiently localized by Sweeps. The worst case computational complexity of Sweeps is exponential. However, extensive experimental evaluations on uniformly random networks modeled by unit disk graphs indicate that Sweeps is practically much more efficient [7]. Part of our future work will be to analyze the average case computational complexity of Sweeps. Additionally, the necessary and sufficient condition for a localizable network to be localizable by Sweeps is that the graph of the network has a bilateration ordering. Extensive simulations on uniformly random networks modeled by unit disk graphs suggest that the gap between localizable and sequentially localizable networks is not large [7]. A question that is of interest is if there exists a threshold such that a graph is globally rigid *and* has a bilateration ordering when the average degree of the graph passes the threshold. In [12], a trilateration-based localization algorithm was proposed for networks with inaccurate distance measurements in which sensors are assigned an estimated position only when the estimated position can be provably bounded to be within some known range of the actual sensor position. A similar concept was employed in adapting the Sweeps algorithm for the case of inaccurate distance measurements [6] in that each estimated sensor position can be guaranteed to be within a known distance of the actual sensor position. As part of future research, we aim to fine tune and improve the Sweeps algorithm adapted for inaccurate distance measurements.

A key aspect of wireless sensor networks is that each sensor can interact with only a subset of the sensors in the network. Hence, Sweeps and Restricted Sweeps are proposed on the assumption that the distances between each sensor and only some of the sensors in a network are known. Although the computations in Sweeps and Restricted Sweeps are currently envisioned as being carried out on a central computer, we note that this does not necessarily contradict the distributed nature of a wireless sensor network. For example, in a sensor network deployed for environment monitoring, quantities measured by a sensor, i.e., chemical emissions, and transmitted to a base station, make sense only in the context of the sensor's position. The distance measurements taken by each sensor to, say, nearby sensors can be transmitted to the base station along with whatever quantities the sensor was deployed to monitor. The base station can then run a localization algorithm using the intersensor distance measurements, and thus associate a position with each measured quantity. An important part of our future research will be to design a fully distributed version of Sweeps.

10. Appendix.

Proof of Lemma 1. Suppose $\mathbb{N}$ is sequentially localizable. For each sensor v, let $k(v)$ denote the sweep in which a finite candidate positions set was computed. Order the sensors as $v_1, \dots, v_n$ so that v_i precedes v_j, i.e., $i < j$, if either $k(v_i) < k(v_j)$ or $k(v_i) = k(v_j)$ and v_i is a predecessor of v_j in the $k(v_i)$th sweep. Consider any v_i

which is not a proxy anchor. First, suppose no distance is known between v_i and any sensor v_j, where $j < i$. This implies that when sensor v_i is processed, there is no known distance between v_i and a sensor whose candidate positions set has already been determined. Hence, there is no data with which to compute a finite candidate positions set for v_i. Now suppose the distance between v_i and exactly one other sensor v_j, $j < i$, is known. This implies that when v_i is processed, its distance to exactly one sensor with a finite candidate positions set is known. By definition of a sequential localization algorithm, a sensor for which a finite candidate positions set has not been computed does not have any position information associated with it. Hence, since sensor positions are distinct, a finite candidate positions set of v_i cannot be determined when just its distance to a single sensor with an already computed candidate positions set is known. We have just shown that if v_i is not a proxy anchor, then v_i must be adjacent to at least two v_j, where $j < i$. This implies that v_1 and v_2 must be proxy anchors, and so $v_1, \dots, v_n$ is a bilateration ordering. □

Proof of Lemma 2. The "if" direction has already been shown in section 3. The "only if" direction is a straightforward consequence of the following. Given a sensor v, and its distances to k sensors $u_1, \dots, u_k$ with known positions where no three sensors in $\{v, u_1, \dots, u_k\}$ are collinear, there exists exactly one position for sensor v such that its distances to all k sensors are satisfied if and only if $k \geq 3$. □

Proof of Lemma 3. Let $\mathcal{V}$ denote the vertex set of $\mathbb{G}$, and let u be any vertex in $\mathcal{V}$. Note that u must be adjacent to at least one other vertex in $\mathbb{G}$ since $\mathbb{G}$ is rigid and therefore connected. Let v be any vertex adjacent to u. Let $u_1, u_2, \dots, u_m$ be any ordering of $m \leq |\mathcal{V}|$ vertices such that $u_1 = u$, $u_2 = v$, and each u_i, $i \geq 3$, is adjacent to at least two vertices u_k, $k < i$. Moreover, suppose there does not exist any vertex $w \in \mathcal{V} - \{u_1, u_2, \dots, u_m\}$ which is adjacent to two or more vertices in $\{u_1, u_2, \dots, u_m\}$. Let $\mathbb{B}$ denote the graph induced in $\mathbb{G}$ by $\{u_1 = u, u_2 = v, \dots, u_m\}$. Note that $\mathbb{B}$ contains at least two vertices, namely, u and v. In the following we will show that $\mathcal{V} - \{u_1, u_2, \dots, u_m\} \neq \emptyset$ is a contradiction to the assumption that $\mathbb{G}$ is chordal. So, suppose $\mathcal{V} - \{u_1, u_2, \dots, u_m\} \neq \emptyset$. Let $\mathbb{F}$ denote a maximally connected component of the graph induced by vertices not in $\mathbb{B}$, i.e., $\mathcal{V} - \{u_1, u_2, \dots, u_m\}$. Note that $\mathbb{F}$ has at least one vertex since $\mathcal{V} - \{u_1, u_2, \dots, u_m\} \neq \emptyset$. For an edge incident on vertices a and b, we say that the edge is *from* $\mathbb{B}$ to $\mathbb{F}$ if a is in $\mathbb{B}$ and b is in $\mathbb{F}$. Since $\mathbb{G}$ is rigid, and $\mathbb{F}$ contains at least one vertex, there must be at least two edges e_1 and e_2 from $\mathbb{B}$ to $\mathbb{F}$. Let $e_1, e_2, \dots, e_c$, $c \geq 2$, denote all the edges from $\mathbb{B}$ to $\mathbb{F}$.

A vertex in $\mathbb{F}$ can be incident on at most one edge from $\mathbb{B}$ to $\mathbb{F}$. For if a vertex w in $\mathbb{F}$ is incident on two edges from $\mathbb{B}$ to $\mathbb{F}$, then obviously, $w \in \mathcal{V} - \{u_1, u_2, \dots, u_m\}$ and w is adjacent to two or more vertices in $\{u_1, u_2, \dots, u_m\}$, which contradicts our assumption that there does not exist any vertex $w \in \mathcal{V} - \{u_1, u_2, \dots, u_m\}$ which is adjacent to two or more vertices in $\{u_1, u_2, \dots, u_m\}$. Suppose e_1 is incident on vertex z in $\mathbb{B}$, and that all the edges from $\mathbb{B}$ to $\mathbb{F}$ are incident on z. By removing z from $\mathbb{G}$, $\mathbb{G}$ is disconnected since this removes all edges from $\mathbb{B}$ to $\mathbb{F}$, and $\mathbb{F}$ is a maximally connected component of the graph induced by vertices not in $\mathbb{B}$. But $\mathbb{G}$ is rigid, which means it is at least two connected, and therefore it requires the removal of at least two vertices to disconnect $\mathbb{G}$. Hence, there must exist at least one edge from $\mathbb{B}$ to $\mathbb{F}$ which is not incident on z. So there must exist two edges from $\mathbb{B}$ to $\mathbb{F}$ such that the edges are incident on distinct vertices in $\mathbb{B}$. Let e_i and e_j denote two such edges. Also from the above, we have that e_i and e_j are incident on distinct vertices in $\mathbb{F}$. Hence, there exist distinct vertices $b, b' \in \mathbb{B}$ and $f, f' \in \mathbb{F}$ such that b is adjacent to f and b' is adjacent to f'. Since $\mathbb{B}$ is connected, there is a path in $\mathbb{B}$ from b to b'. Let this path be denoted $b_0 = b, b_1, b_2, \dots, b_B = b'$. Since $\mathbb{F}$ is connected, there is a path in $\mathbb{F}$

from f to f'. Let this path be denoted $f_0 = f, f_1, f_2, \ldots, f_F = f'$.

Let L be the smallest positive integer in $\{1, 2, \ldots, F\}$ such that f_L is adjacent to some vertex in $\{b_1, b_2, \ldots, b_B\}$. Note that such an L must exist since $f' = f_F$ is adjacent to $b' = b_B$. Let $\bar{L}$ be such that $b_{\bar{L}}$ is the vertex in $\{b_1, b_2, \ldots, b_B\}$ to which f_L is adjacent. Note that $\bar{L} > 0$ and $L > 0$. Let T be the largest integer less than L such that f_T is adjacent to some vertex in $\{b_0, b_1, \ldots, b_{\bar{L}-1}\}$. Such a T must exist since $f = f_0$ is adjacent to $b = b_0$, and as noted above, $L, \bar{L} > 0$. Let $\bar{T}$ be such that $b_{\bar{T}}$ is the vertex to which f_T is adjacent in $\{b_0, b_1, \ldots, b_{\bar{L}-1}\}$. By construction, the subgraph of $\mathbb{G}$ with vertices $f_T, f_{T+1}, \ldots, f_L, b_{\bar{L}}, b_{\bar{L}-1}, \ldots, b_{\bar{T}}$ and edges $(f_T, f_{T+1}), \ldots, (f_{L-1}, f_L), (f_L, b_{\bar{L}}), (b_{\bar{L}-1}, b_{\bar{L}-2}), \ldots, (b_{\bar{T}-1}, b_{\bar{T}}), (b_{\bar{T}}, f_T)$ is a cycle. Let this cycle be denoted $\mathbb{C}$. Note that $\mathbb{C}$ contains at least four vertices and so is a cycle of length at least four. Since $\mathbb{G}$ is chordal, there must exist an edge in $\mathbb{G}$ that is also a chord of $\mathbb{C}$. Since f_T and f_L can each be incident upon only one edge from $\mathbb{B}$ to $\mathbb{F}$, we have that any chord of $\mathbb{C}$ that is also an edge from $\mathbb{B}$ to $\mathbb{F}$ must be incident upon f_M, where $T < M < L$. Suppose there is such a vertex f_M. Since L is the smallest positive integer in $\{1, 2, \ldots, F\}$ such that f_L is adjacent to some vertex in $\{b_1, b_2, \ldots, b_B\}$, it follows that f_M must be adjacent to b_0 since $M < L$. But this is a contradiction since $M > T$ and T is the largest integer less than L such that f_T is adjacent to some vertex in $\{b_0, b_1, \ldots, b_{\bar{L}-1}\}$. Hence, any chord of $\mathbb{C}$ can only contain vertices which are either both in $\mathbb{B}$ or both in $\mathbb{F}$. Since $\mathbb{C}$ contains at least four vertices, two of which are in $\mathbb{B}$ and two of which are in $\mathbb{F}$, and $\mathbb{C}$ contains no chord from $\mathbb{B}$ to $\mathbb{F}$, it follows that there is a chordless cycle in $\mathbb{G}$ of at least four vertices. This contradicts the fact that $\mathbb{G}$ is chordal. Hence, it cannot be the case that $\mathcal{V} - \{u_1, u_2, \ldots, u_m\} \neq \emptyset$. So, $\mathbb{B}$ contains all the vertices in $\mathcal{V}$, which implies $\mathbb{G}$ has a bilateration ordering. Recall that vertices u and v of $\mathbb{B}$, which are the first two vertices of the bilateration ordering on all the vertices of $\mathbb{B}$, may be any two vertices of $\mathbb{G}$. As shown above, $\mathbb{B}$ must contain all the vertices of $\mathbb{G}$. Hence, it must be the case that for all edges (u, v) in $\mathbb{G}$, there exists a bilateration ordering of $\mathbb{G}$ that begins with vertices u and v. □

Proof of Lemma 4. If w is a proxy anchor, then the lemma holds trivially, so suppose w is not a proxy anchor. It is also easy to show that the lemma holds when u and v are both proxy anchors, so suppose at least one of u or v is not a proxy anchor. Let the ordering of the first sweep be $x_1, \ldots, x_n$, which we denote by $[x]$. Let $w = x_k$. Without loss of generality, suppose u precedes v in $[x]$, i.e., $u = x_i$ and $v = x_j$ for some i, j where $i < j$. It is easy to see that if w precedes v in the ordering $[x]$, then it cannot be the case that v is in the domain of any subassignment in $\mathcal{S}(w, 1)$. Therefore, it must be the case that $k \geq j$. We will prove the lemma by induction on $k - j$. First, consider the case where $k = j$. In this case, $w = v$. From (13), it is clear that $u \in \mathcal{D}(\beta)$ for all $\beta \in \mathcal{S}(v, 1)$, and $\|\beta(u) - \beta(v)\| = d_{uv}$. Now suppose $k - j = 1$. Let $\mathcal{M}(w) = \mathcal{N}(w) \cap \{x_1, \ldots, x_{k-1}\}$. If $x_k = w$ is not adjacent to $x_j = v$, then it is easy to see that v cannot be in the domain of any subassignment in $\mathcal{S}(w, 1)$. Hence, suppose w is adjacent to v, which implies $v \in \mathcal{M}(w)$. Let the elements of $\mathcal{M}(w)$ be denoted $u_1, \ldots, u_m$, and without loss of generality, let $v = u_1$. Let β be any subassignment of $\mathcal{S}(w, 1)$. From (15), we have that $\beta \in \mathcal{M}(\alpha_1, \ldots, \alpha_m, w, u_1, \ldots, u_m)$ for some collection of pairwise consistent subassignments $\alpha_1, \ldots, \alpha_m$ where $\alpha_i \in \mathcal{S}(u_i, 1)$, $i \in \{1, \ldots, m\}$. This implies $\mathcal{D}(\alpha_1) \subset \mathcal{D}(\beta)$. As we have just shown, $u \in \mathcal{D}(\delta)$ and $\|\delta(u) - \delta(v)\| = d_{uv}$ for all $\delta \in \mathcal{S}(v, 1)$. Hence, $u, v \in \mathcal{D}(\alpha_1)$ and $\|\alpha_1(u) - \alpha_1(v)\| = d_{uv}$. From (13), we have that the domain of each subassignment in $\mathcal{M}(\alpha_1, \ldots, \alpha_m, w, u_1, \ldots, u_m)$ contains $\mathcal{D}(\alpha_1)$ and must be identical to α_1 when restricted to the domain of α_1. Therefore, $u, v \in \mathcal{D}(\beta)$ and

$\beta(u) = \alpha_1(u)$, $\beta(v) = \alpha_1(v)$. Clearly, this implies $\|\beta(u) - \beta(v)\| = d_{uv}$. Hence, for all $\gamma \in \mathcal{S}(w, 1)$ we have that $u, v \in \mathcal{D}(\gamma)$ and $\|\gamma(u) - \gamma(v)\| = d_{uv}$.

Suppose the lemma holds for all $w = x_k$ where $k - j \leq L$ for some L. Now consider w, where $w = x_k$, where $k - j = L + 1$. Again, let $u_1, \ldots, u_m$ denote the elements of $\mathcal{M}(w)$. Let β be any subassignment of $\mathcal{S}(w, 1)$ where $u, v \in \mathcal{D}(\beta)$. From (15), we have that $\beta \in \mathcal{M}(\alpha_1, \ldots, \alpha_m, w, u_1, \ldots, u_m)$ for some collection of pairwise consistent subassignments $\alpha_1, \ldots, \alpha_m$ where $\alpha_i \in \mathcal{S}(u_i, 1)$, $i \in \{1, \ldots, m\}$. By definition, the domain of β is equal to the union of $\{w\}$ and the union of the domains of α_z, $z \in \{1, \ldots, m\}$. Hence, it must be the case that $v \in \mathcal{D}(\alpha_z)$ for some $z \in \{1, \ldots, m\}$. As noted previously, u is in the domain of all subassignments in $\mathcal{S}(v, 1)$. It is straightforward to show then that $u, v \in \mathcal{D}(\alpha_z)$. By definition of $\mathcal{M}$ in (13), we have that β must equal α_z when restricted to the domain of α_z. This implies $\beta(u) = \alpha_z(u)$ and $\beta(v) = \alpha_z(v)$. By the inductive hypothesis, we have that $\|\alpha_z(u) - \alpha_z(v)\| = d_{uv}$, which implies $\|\beta(u) - \beta(v)\| = d_{uv}$. The lemma follows by induction. □

Proof of Lemma 5. It is easy to show that the lemma holds when u and v are both proxy anchors, so suppose at least of of u or v is not a proxy anchor. Let $y_1, \ldots, y_n$ denote the ordering chosen for the second sweep, and for any vertex y_i, let $\mathcal{M}(y_i) = \mathcal{N}(y_i) \cap \{y_1, \ldots, y_{i-1}\}$. Suppose $w = y_i$. We will prove the lemma by induction on $i \in \{1, \ldots, n\}$. The lemma is trivially true if w is a proxy anchor, i.e., $w = y_i$, $i \in \{1, 2, 3\}$. Now suppose the lemma holds for all $w = y_j$, where $j < i$ for some $i \in \{4, \ldots, n\}$. Consider $w = y_i$. If $\mathcal{M}(w) = \emptyset$, then $\mathcal{S}(w, 2) = \mathcal{S}(w, 1)$, in which case Lemma 5 follows from Lemma 4. So suppose $\mathcal{M}(w) \neq \emptyset$. Let $u_1, \ldots, u_m$ denote the elements of $\mathcal{M}(w)$. Let β be any subassignment of $\mathcal{S}(w, 2)$ such that $u, v \in \mathcal{D}(\beta)$. From (17), we have that $\beta = u_{m+1}(\alpha_0, \alpha_1, \ldots, \alpha_m)$ for some collection of pairwise consistent subassignments $\alpha_0 \in \mathcal{S}(w, 1)$ and $\alpha_j \in \mathcal{S}(u_j, 2)$, $j \in \{1, \ldots, m\}$. Without loss of generality, suppose that u preceded v in the ordering chosen for the first sweep. By definition of u_{m+1}, the domain of β is the union of the domains of α_j, $j \in \{0, 1, \ldots, m\}$. Hence, since $u, v \in \mathcal{D}(\beta)$, it follows that $v \in \mathcal{D}(\alpha_z)$ for some $z \in \{0, 1, \ldots, m\}$. It is straightforward to show from (17) that u must also be in $\mathcal{D}(\alpha_z)$, so $u, v \in \mathcal{D}(\alpha_z)$. From Lemma 4 and the inductive hypothesis, it follows that $\|\alpha_z(u) - \alpha_z(v)\| = d_{uv}$. Since α_j, $j \in \{0, 1, \ldots, m\}$, are pairwise consistent, it follows from the definition of u_{m+1} that $\beta(u) = \alpha_z(u)$ and $\beta(v) = \alpha_z(v)$, so $\|\beta(u) - \beta(v)\| = d_{uv}$. □

Proof of Lemma 6. We first show that the lemma holds for the first sweep, i.e., for each $v \in \mathcal{V}$, there is a $\beta \in \mathcal{S}(v, 1)$ which is the restriction of $\bar{\alpha}$ to $\mathcal{D}(\beta)$, and $v \in \mathcal{D}(\beta)$. Let $v_1, \ldots, v_n$ be the ordering used to compute $\mathcal{S}(v, 1)$, $v \in \mathcal{V}$. Let v be any vertex of $\mathbb{G}$. If v is a proxy anchor, i.e., $v = v_i$ for some $i \in \{1, 2, 3\}$, then from (14), we have that $\mathcal{S}(v, 1) = \{\alpha\}$, where $\mathcal{D}(\alpha) = \{v\}$ and $\alpha(v) = \pi(v)$. Hence, the lemma holds for v_i, $i \in \{1, 2, 3\}$. Now suppose $v = v_i$, $i > 3$. From (13) and (15), we have that $v_i \in \mathcal{D}(\beta)$ for all $\beta \in \mathcal{S}(v, 1)$, so it just remains to show that there is a $\beta \in \mathcal{S}(v, 1)$ which is the restriction of $\bar{\alpha}$ to $\mathcal{D}(\beta)$. We show this by induction on v_i, $i \in \{1, 2, \ldots, n\}$. We have already shown the lemma to be true for $\mathcal{S}(v_i, 1)$, $i \in \{1, 2, 3\}$. Now suppose the lemma holds for all $\mathcal{S}(v_j, 1)$, where $j < i$ for some $i \in \{4, \ldots, n\}$. For each v_j, $j < i$, let $\bar{\beta}_{v_j}$ denote the subassignment in $\mathcal{S}(v_j, 1)$ which is the restriction of $\bar{\alpha}$ to $\mathcal{D}(\bar{\beta}_{v_j})$. Now consider v_i. Let the elements of $\mathcal{N}(v_i) \cap \{v_1, \ldots, v_{i-1}\}$ be denoted $u_1, \ldots, u_m$. Clearly, $\bar{\beta}_{u_j}$, $j \in \{1, \ldots, m\}$, are pairwise consistent, i.e., $\bar{\beta}_{u_j} \sim \bar{\beta}_{u'_j}$ for all $j, j' \in \{1, \ldots, m\}$. Consider $\mathcal{M}(\bar{\beta}_{u_1}, \ldots, \bar{\beta}_{u_m}, v_i, u_1, \ldots, u_m)$. Clearly, $\pi(v_i) \in \bigcap_{j \in \{1, \ldots, m\}} \mathcal{C}(\bar{\beta}_{u_j}(u_j), d_{v_i u_j})$. From (13), it follows that there is $\beta \in \mathcal{S}(v_i, 1)$ such that $\beta = \bar{\beta}_{u_j}$ when restricted to the domain of $\bar{\beta}_{u_j}$ for each $j \in \{1, \dots, m\}$, and $\beta(v_i) = \pi(v_i)$. Since each $\bar{\beta}_{u_j}$ is a restriction of $\bar{\alpha}$, it follows that β must then be a restriction of $\bar{\alpha}$ as well. By induction then, we have that the lemma is true for all $\mathcal{S}(v, 1)$, $v \in \mathcal{V}$.

We have just shown that the lemma holds for the first sweep. Now we will show the lemma holds for all sweeps by induction, so suppose the lemma holds for the kth sweep, where $k \geq 1$. Let $u_1, \dots, u_n$ be the ordering chosen for the $(k+1)$th sweep. Clearly, $\mathcal{S}(u_i, k+1) = \mathcal{S}(u_i, k)$ for $i \in \{1, 2, 3\}$, and since the lemma holds for the kth sweep, we have that the lemma holds for $\mathcal{S}(u_i, k+1)$, $i \in \{1, 2, 3\}$. Suppose the lemma holds for all $\mathcal{S}(u_j, k+1)$, where $j < i$ for some $i \in \{4, \dots, n\}$. For each u_j, $j < i$, let $\bar{\beta}_{u_j}$ denote the subassignment in $\mathcal{S}(u_j, k+1)$ which is a restriction of $\bar{\alpha}$ with u_j in its domain. Consider u_i. Clearly, if $\mathcal{N}(u_i) \cap \{u_1, \dots, u_{i-1}\}$ is the empty set, then $\mathcal{S}(u_i, k+1) = \mathcal{S}(u_i, k)$, in which case the lemma holds for $\mathcal{S}(u_i, k+1)$ since the lemma is true for the kth sweep. So suppose $\mathcal{N}(u_i) \cap \{u_1, \dots, u_{i-1}\} \neq \emptyset$, and let $w_1, \dots, w_m$ denote its elements. By the inductive hypothesis, we have subassignments $\bar{\beta}_{u_i} \in \mathcal{S}(u_i, k)$ and $\bar{\beta}_{w_j} \in \mathcal{S}(w_j, k+1)$, $j \in \{1, \dots, m\}$, where each subassignment is a restriction of $\bar{\alpha}$ and $u_i \in \mathcal{D}(\bar{\beta}_{u_i})$, $w_j \in \mathcal{D}(\bar{\beta}_{w_j})$, $j \in \{1, \dots, m\}$. From (17), it is easy to see that $u_{m+1}(\bar{\beta}_{u_i}, \bar{\beta}_{w_1}, \dots, \bar{\beta}_{w_m})$ is in $\mathcal{S}(u_i, k+1)$ and is also a restriction of $\bar{\alpha}$ with u_i in its domain. By induction, the lemma holds for the $(k+1)$th sweep. ☐

Proof of Lemma 7. Let $\mathbb{H}_i'$ denote the graph induced in $\mathbb{G}$ by the vertices of $\mathbb{H}_i$ and the vertices of $\mathcal{A}$. Suppose $\mathbb{H}_i'$ is not globally rigid. Consider the subnetwork of $\mathbb{N}$ containing just the sensors corresponding to vertices in $\mathbb{H}_i'$, and denote the subnetwork by $\mathbb{N}_i$. Clearly, the point formation modeling $\mathbb{N}_i$ is $(\mathbb{H}_i', p')$, where p' contains the positions of those sensors of $\mathbb{N}_i$. Since the multipoint of $\mathbb{N}$ is generic, it follows that the multipoint of $\mathbb{N}_i$, i.e., p', must also be generic. Hence, that $\mathbb{H}_i'$ is not globally rigid implies $\mathbb{N}_i$ cannot be localizable. In other words, there exists multipoint q' such that the point formations $(\mathbb{H}_i', q')$ and $(\mathbb{H}_i', p')$ have the same edge lengths but are not congruent. Furthermore, it is easy to see that by applying a Euclidean transformation to the points of $(\mathbb{H}_i', q')$, we can obtain a point formation $(\mathbb{H}_i', q'')$, which is congruent to $(\mathbb{H}_i', q')$, and such that the points in q'' corresponding to the vertices in $\mathcal{A}$ are identical to the points in p' corresponding to the vertices in $\mathcal{A}$. Hence, $(\mathbb{H}_i', q'')$ has the same edge lengths as $(\mathbb{H}_i', p')$, and the points corresponding to vertices in $\mathcal{A}$ are the same in both $(\mathbb{H}_i', q'')$ and $(\mathbb{H}_i', p')$, but $(\mathbb{H}_i', q'')$ and $(\mathbb{H}_i', p')$ are not congruent. Let $(\mathbb{G}, p)$ be the point formation modeling $\mathbb{N}$. Since $\mathbb{N}$ is localizable, it follows that $(\mathbb{G}, p)$ is globally rigid. Consider the point formation $(\mathbb{G}, p'')$ defined as follows. The point in $(\mathbb{G}, p'')$ corresponding to a vertex j not in $\mathbb{H}_i$ is the same as the point corresponding to vertex j in $(\mathbb{G}, p)$, and the point in $(\mathbb{G}, p'')$ corresponding to a vertex j in $\mathbb{H}_i$ is the same as the point corresponding to vertex j in $(\mathbb{H}_i', q'')$. It is easy to see that $(\mathbb{G}, p'')$ has the same edge lengths as $(\mathbb{G}, p)$ but $(\mathbb{G}, p'')$ and $(\mathbb{G}, p)$ are not congruent. This contradicts the fact that $(\mathbb{G}, p)$ is globally rigid, and therefore $\mathbb{H}_i'$ must be globally rigid. ☐

Proof of Lemma 8. In the following, we will show that the lemma holds for the case where $c = 1$; i.e., the graph induced in $\mathbb{G}$ by vertices not in $\mathcal{A}$ is connected. The case for $c > 1$ follows easily. Let $\mathbb{H}$ denote the graph induced in $\mathbb{G}$ by vertices not in $\mathcal{A}$. Let the ordering for the second sweep be $[x] = x_1, \dots, x_n$, and suppose $[x]$ is a complete ordering of $\mathbb{H}$ with respect to v of $\mathbb{H}$ and $\mathcal{A}$. This implies $v = x_n$, and $[x]$ is also a complete ordering of $\mathbb{G}$ with respect to $\mathcal{A}$ and x_n.

Let x_i be any vertex such that there is a path from x_i to x_n in $\mathbb{G}$ which is a subsequence of $[x]$ beginning with a vertex which is not a proxy anchor. In other words, there exists $i < i_1 < i_2 < \cdots < i_p < n$ such that $i_1 > 3$ and $(x_i, x_{i_1}), (x_{i_1}, x_{i_2}),$

$\dots,(x_{i_p},x_n)\in\mathcal{E}$. We will show by induction that $x_i\in\mathcal{D}(\beta)$ for all $\beta\in\mathcal{S}(x_n,2)$. For notational convenience, let $i_{p+1}=n$. For any x_j, let $\mathcal{M}(x_j)=\mathcal{N}(x_j)\cap\{x_1,\dots,x_{j-1}\}$. Clearly, $x_i\in\mathcal{M}(x_{i_1})$. From (17), it follows that $x_i\in\mathcal{D}(\beta)$ for all $\beta\in\mathcal{S}(x_{i_1},2)$. Now suppose $x_i\in\mathcal{D}(\beta)$ for all $\beta\in\mathcal{S}(x_{i_j},2)$, where $j\le I$ for some $I<p+1$, and consider $\mathcal{S}(x_{i_{j+1}},2)$. Since $x_{i_j}\in\mathcal{M}(x_{i_{j+1}})$, it follows from (17) that for all $\beta\in\mathcal{S}(x_{i_{j+1}},2)$, it must be the case that $\mathcal{D}(\beta')\subseteq\mathcal{D}(\beta)$, where β' is some subassignment of $\mathcal{S}(x_{i_j},2)$. But since $x_i\in\mathcal{D}(\beta')$ for all $\beta'\in\mathcal{S}(x_{i_j},2)$, it follows that $x_i\in\mathcal{D}(\beta)$ for all $\beta\in\mathcal{S}(x_{i_{j+1}},2)$. By induction then, we have that $x_i\in\mathcal{D}(\beta)$ for all $\beta\in\mathcal{S}(x_n,2)$.

Let $\mathcal{N}_0(x_n),\dots,\mathcal{N}_h(x_n)$ be the vertex partition of $\mathbb{H}$ used to construct the complete ordering $[x]$. Consider any x_i, $i>3$, and suppose $x_i\in\mathcal{N}_j(x_n)$. Now we show that there is a path from x_i to x_n in $\mathbb{G}$ which is a subsequence of $[x]$; i.e., there exists $i<i_1<i_2<\cdots<i_p<n$ such that $(x_i,x_{i_1}),(x_{i_1},x_{i_2}),\dots,(x_{i_p},x_n)\in\mathcal{E}$. Since $x_i\in\mathcal{N}_j(x_n)$, it must be true that x_i is adjacent to some vertex in $\mathcal{N}_{j'}(x_n)$, where $j'<j$. But since all the vertices in sets $\mathcal{N}_{j'}$, $j'<j$, are assigned larger indices than vertices in $\mathcal{N}_j$, it follows that x_i must be adjacent to some vertex x_{i_1} where $i<i_1$. If $x_{i_1}\in\mathcal{N}_0$, then it must be the case that $x_n=x_{i_1}$. Otherwise, $x_{i_1}\in\mathcal{N}_b$ where $b>0$, and so x_{i_1} must be adjacent to some vertex x_{i_2} in $\mathcal{N}_a$ where $a<b$. By construction of a complete ordering, we have that $i_1<i_2$ since $x_{i_2}\in\mathcal{N}_a$ and $a<b$. Hence, there must exist a sequence of vertices $x_{i_1},\dots,x_{i_p}=x_n$ such that x_i is adjacent to x_{i_1}, each x_{i_j} is adjacent to $x_{i_{j+1}}$, and $i<i_1<i_2<\cdots<i_p=n$.

From the above, we can conclude that each x_i, $i>3$, must be in $\mathcal{D}(\beta)$ for all $\beta\in\mathcal{S}(x_n,2)$. Now consider the proxy anchors, i.e., x_i, $i\in\{1,2,3\}$. Suppose some x_i, $i\in\{1,2,3\}$, is not adjacent to any x_j, $j>3$. This implies x_i has degree two in $\mathbb{G}$, and therefore $\mathbb{G}$ cannot be globally rigid in $\mathbb{R}^2$, and $\mathbb{N}$ is not localizable. This is clearly a contradiction. Hence, each x_i, $i\in\{1,2,3\}$, must be adjacent to some x_j, $j>3$, which implies there exist indices $i_1,\dots,i_p$, where $i<i_1<i_2<\cdots<i_p<n$, $i_1>3$, and $(x_i,x_{i_1}),(x_{i_1},x_{i_2}),\dots,(x_{i_p},x_n)\in\mathcal{E}$. Hence, each x_i, $i\in\{1,2,3\}$, must also be in $\mathcal{D}(\beta)$ for all $\beta\in\mathcal{S}(x_n,2)$, and it follows that $\mathcal{D}(\beta)=\mathcal{V}$ for all $\beta\in\mathcal{S}(x_n,2)$. □

Proof of Lemma 9. First, suppose $c=1$ so the graph $\mathbb{H}$ induced in $\mathbb{G}$ by vertices which do not correspond to the proxy anchors is connected. Suppose the ordering used to compute the second sweep is a complete ordering of $\mathbb{G}$ with respect to $\mathcal{A}$ and vertex u of $\mathbb{H}$. From Lemma 6, we have that $\mathcal{S}(u,2)$ is not empty. From Lemma 8, we have that for each $\alpha\in\mathcal{S}(u,2)$, the domain of α is equal to $\mathcal{V}$. From Lemma 5 we have that $\|\alpha(u)-\alpha(v)\|=d_{uv}$ for all $(u,v)\in\mathcal{E}$. Clearly, $\alpha(a)=\pi(a)$ for all proxy anchors a. Hence, α is a consistent assignment of $\mathbb{N}$, where $\alpha(a)=\bar{\alpha}(a)$ for all proxy anchors a. But as noted previously, there can be at most one such assignment, which implies that α must equal $\bar{\alpha}$. Now we consider the case for $c>1$. It follows from Lemma 7 that each subnetwork $\mathbb{N}_i$ containing sensors corresponding to vertices in $\mathcal{A}$ and $\mathbb{H}_i$ is itself localizable. The argument for the case $c=1$ can be applied to each $\mathbb{N}_i$ to show that each subassignment in $\mathcal{S}(u_i,2)$ must be the restriction of $\bar{\alpha}$ to the vertices corresponding to the proxy anchors and the sensors in $\mathbb{N}_i$. □

Proof of Theorem 2. We prove the lemma for the case where $c=1$; i.e., the graph $\mathbb{G}(\mathcal{V}-\mathcal{W})$ is connected. The lemma for the case where $c>1$ is a direct consequence.

Since $v_1,\dots,v_n$ is the ordering chosen for the first sweep, without loss of generality we can suppose v_1,v_2,v_3 are the proxy anchors and let $\mathcal{W}=\{v_1,v_2,v_3\}$. Since $\mathbb{G}$ is partially acyclic and $c=1$, it follows that $\mathbb{G}(\mathcal{V}-\mathcal{W})$ is acyclic and connected. Hence, for each $v\in\mathcal{V}-\mathcal{W}$, we can define $l(v)$, where $l(v)$ is the length of the path from v to v_n in $\mathbb{G}(\mathcal{V}-\mathcal{W})$. Clearly $l(v_n)=0$. Let $L=\max_{v\in\mathcal{V}-\mathcal{W}}l(v)$. For each v_i, let $\mathcal{M}(v_i)=\mathcal{N}(v_i)\cap\{v_1,\dots,v_{i-1}\}$. Let $[v]$ denote the ordering $v_1,\dots,v_n$. Let p be any point in $\mathcal{S}(v_n,1)$. In the following we will assign point $p(v)$ to each sensor $v\in\mathcal{V}-\mathcal{W}$ so that all known intersensor distances are satisfied and the point assigned to v is p, i.e., $p(v_n) = p$. We do this inductively on $l(v)$ beginning with v where $l(v) = 0$. Obviously v_n is the only vertex such that $l(v_n) = 0$, and we let $p(v_n) = p$. Now we consider v where $l(v) = 1$. As noted previously, $\mathcal{M}(v_n) - \mathcal{W} = \mathcal{N}_1(v_n)$, and by definition $\mathcal{N}_1(v_n)$ is the set of vertices v where $l(v) = 1$. Let $u_1, \dots, u_m$ denote the vertices in $\mathcal{M}(v_n) - \mathcal{W}$. From (2), there are points $p_i \in \mathcal{S}(u_i, 1)$ such that $\|p - p_i\| = d_{v_n u_i}$. For each u_i, $i \in \{1, \dots, m\}$, let $p(u_i) = p_i$. Now suppose $p(v)$ has been defined for all vertices v where $l(v) \leq k$ for some $k < L$. Now we define $p(v)$ for each vertex $v \in \mathcal{V} - \mathcal{W}$ where $l(v) = k + 1$. Since $[v]$ is a complete ordering and $\mathbb{G}(\mathcal{V} - \mathcal{W})$ is acyclic, we have that if vertex v is such that $l(v) = k+1$, then there must exist exactly one vertex $v' \in \mathcal{V} - \mathcal{W}$ such that $(v, v') \in \mathcal{E}$, and $l(v') \leq k$. Since $l(v') = k$, it follows that $p(v')$ has already been defined. Furthermore, from (2), we have that there must exist point $p_v \in \mathcal{S}(v, 1)$ such that $\|p(v') - p_v\| = d_{vv'}$. Let $p(v) = p_v$.

Let u, v be any two vertices in $\mathcal{V} - \mathcal{W}$ which are adjacent in $\mathbb{G}$. Since $\mathbb{G}(\mathcal{V} - \mathcal{W})$ is acyclic, it follows that either $l(u) = l(v) + 1$ or $l(v) = l(u) + 1$, which implies $\|p(u) - p(v)\| = d_{uv}$. Now let w be any vertex of $\mathcal{W}$, and let u be any vertex of $\mathcal{V} - \mathcal{W}$. Since all the sensors in $\mathcal{W}$ are proxy anchors, each $w \in \mathcal{W}$ is assigned some position $\pi(w)$ by the first sweep of the Restricted Sweeps algorithm. Again from (2), we have that for all $p_u \in \mathcal{S}(u, 1)$, it must be the case that $\|p(u) - \pi(w)\| = d_{uw}$. Hence, if we define $p(w) = \pi(w)$ for all $w \in \mathcal{W}$, and assigned position $p(v)$ to each sensor $v \in \mathcal{V}$, then all known intersensor distances must be satisfied. We have just shown that for each $p \in \mathcal{S}(v_n, 1)$, there correspond points $p(v)$, $v \in \mathcal{V}$, such that $p(v_n) = p$ and $p(w) = \pi(w)$ for all $w \in \mathcal{W}$, and all known intersensor distances are satisfied. Since $\mathbb{N}$ is localizable, we have that for all assignments of points $q(v)$ to sensors $v \in \mathcal{V} - \mathcal{W}$ such that all known intersensor distances are satisfied, assuming each sensor $w \in \mathcal{W}$ is positioned at $\pi(w)$, it must be the case that $q(v) = p(v)$ for all $v \in \mathcal{V} - \mathcal{W}$. This implies $\mathcal{S}(v_n, 1)$ can contain only one element. Let $\pi(v_n)$ denote the point in $\mathcal{S}(v_n, 1)$. Clearly, $\pi(v_n)$ is the position of sensor v_n relative to the positions assigned to the proxy anchors. Let $p(v)$, $v \in \mathcal{V}$, be as defined above. Now we show that $\mathcal{S}(v, 2)$, $v \in \mathcal{V}$, must all be singletons. This is trivially true for $v \in \mathcal{W} \cup \{v_n\}$, so consider $v \notin \mathcal{W} \cup \{v_n\}$. Since $\mathbb{G}(\mathcal{V} - \mathcal{W})$ is acyclic, it follows that v is adjacent to exactly one v' in $\mathbb{G}(\mathcal{V} - \mathcal{W})$ such that $l(v') < l(v)$. By definition, $\|p(v) - p(v')\| = d_{vv'}$, and the only criterion used for choosing $p(v)$ from $\mathcal{S}(v, 1)$ was that $\|p(v) - p(v')\| = d_{vv'}$. This implies that if there is $q \in \mathcal{S}(v, 1)$ where $q \neq p(v)$ and $\|q - p(v')\| = d_{vv'}$, then there exists an assignment of points $q(x)$, $x \in \mathcal{V}$, such that $q(v) = q$, $q(w) = \pi(w)$ for all $w \in \mathcal{W}$, and all known intersensor distances are satisfied. But this clearly contradicts the assumption that $\mathbb{N}$ is localizable. Hence, there can exist only one point $p \in \mathcal{S}(v, 1)$, namely $p(v)$, such that $\|p - p(v')\| = d_{vv'}$. From (3) and the ordering specified for the second sweep, it follows that $\mathcal{S}(v, 2)$ must be a singleton consisting of only $p(v)$. Hence, $\mathcal{S}(v, 2)$ must be a singleton for all $v \in \mathcal{V}$. □

Proof of Lemma 13. Without loss of generality, we assume that v_1, v_2, and v_3 are anchors. Note that the coordinates computed accordingly for the remaining sensors can be transformed into their real locations by Euclidean transformations since there are three anchors in the network. Now consider the first sweep with the ordering $v_1, v_2, \dots, v_n$ in $\bar{\mathbb{G}}^2$. Since v_4 is adjacent to both v_2 and v_3, we have that $\mathcal{S}(v_4, 1)$ contains two elements. Since v_5 is adjacent to both v_4 and v_3, we have that $\mathcal{S}(v_5, 1)$ contains four elements. Similarly, $\mathcal{S}(v_i, 1)$, $4 \leq i \leq n$ contains finite elements by using the edges (v_i, v_{i-1}) and (v_i, v_{i-2}) in $\bar{\mathbb{G}}^2$. Then consider the second sweep with the ordering $v_1, v_2, v_3, v_n, v_{n-1}, \dots, v_4$. From Lemma 2.1 in [14] we know that $\bar{\mathbb{G}}^2$ is generically globally rigid, which implies that generically there is one element in

$\mathcal{S}(v_n, 1)$ which satisfies simultaneously $||p_{v_n} - p_{v_1}|| = d_{v_n v_1}$ and $||p_{v_n} - p_{v_2}|| = d_{v_n v_2}$. Hence, $\mathcal{S}(v_n, 2)$ contains exactly one element. Using the same reasoning, we know that $\mathcal{S}(v_{n-1}, 2)$ contains one element by using the edges (v_{n-1}, v_n) and (v_{n-1}, v_1) in $\bar{\mathbb{G}}^2$. Similarly, we know that $\mathcal{S}(v_i, 2)$, $4 \leq i \leq n-2$ by using the edges (v_i, v_{n+1}) and (v_i, v_{i+2}) in $\bar{\mathbb{G}}^2$. □

REFERENCES

[1] B. D. O. Anderson, P. N. Belhumeur, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, and Y. R. Yang, *Graphical properties of easily localizable sensor networks*, Wireless Networks, 2007 (electronic).

[2] J. Aspnes, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, Y. R. Yang, B. D. O. Anderson, and P. N. Belhumeur, *A theory of network localization*, IEEE Trans. Mobile Computing, 5 (2006), pp. 1663–1678.

[3] P. Biswas, T.-C. Lian, T.-C. Wang, and Y. Ye, *Semidefinite programming based algorithms for sensor network localization*, ACM Trans. Sensor Networks, 2 (2006), pp. 188–220.

[4] R. Connelly, *Generic global rigidity*, Discrete Comput. Geom., 33 (2005), pp. 549–563.

[5] T. Eren, D. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Morse, B. D. O. Anderson, and P. N. Belhumeur, *Rigidity, computation, and randomization in network localization*, in Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), New Haven, CT, 2004, pp. 2673–2684.

[6] J. Fang, D. Duncan, and A. S. Morse, *Sequential localization with inaccurate measurements*, to appear in Localization Algorithms and Strategies for Wireless Sensor Networks: Monitoring and Surveillance Techniques for Target Tracking.

[7] D. K. Goldenberg, P. Bihler, M. Cao, J. Fang, B. D. O. Anderson, A. S. Morse, and Y. R. Yang, *Localization in sparse networks using sweeps*, in Proceedings of Mobicom, 12th Annual International Conference on Mobile Computing and Networking, ACM, New York, 2006, pp. 110–121.

[8] J. E. Graver, B. Servatius, and H. Servatius, *Combinatorial Rigidity*, Grad. Stud. Math. 2, AMS, Providence, RI, 1993.

[9] B. Hendrickson, *Conditions for unique graph realizations*, SIAM J. Comput., 21 (1992), pp. 65–84.

[10] B. Jackson and T. Jordán, *Connected rigidity matroids and unique realizations of graphs*, J. Combin. Theory Ser. B, 94 (2005), pp. 1–29.

[11] D. J. Jacobs and B. Hendrickson, *An algorithm for two-dimensional rigidity percolation: The pebble game*, J. Comput. Phys., 137 (1997), pp. 346–365.

[12] D. Moore, J. Leonard, D. Rus, and S. Teller, *Robust distributed network localization with noisy range measurements*, in Proceedings of the 2nd ACM SenSys, Baltimore, MD, 2004, pp. 50–61.

[13] N. B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, *Poster abstract: Anchor-free distributed localization in sensor networks*, in SenSys '03: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, ACM, New York, 2003, pp. 340–341.

[14] A. Savvides, H. Park, and M. Srivastava, *The n-hop multilateration primitive for node localization problems*, ACM Mobile Networks Appl., 8 (2003), pp. 443–451.

[15] Y. Shang and W. Ruml, *Improved MDS-based localization*, in Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), New Haven, CT, 2004, pp. 2640–2651.

[16] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, *Localization from mere connectivity*, in MobiHoc '03: Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing, ACM, New York, 2003, pp. 201–212.

[17] T. Tay and W. Whiteley, *Generating isostatic frameworks*, Structural Topology, 306 (1988), pp. 115–139.